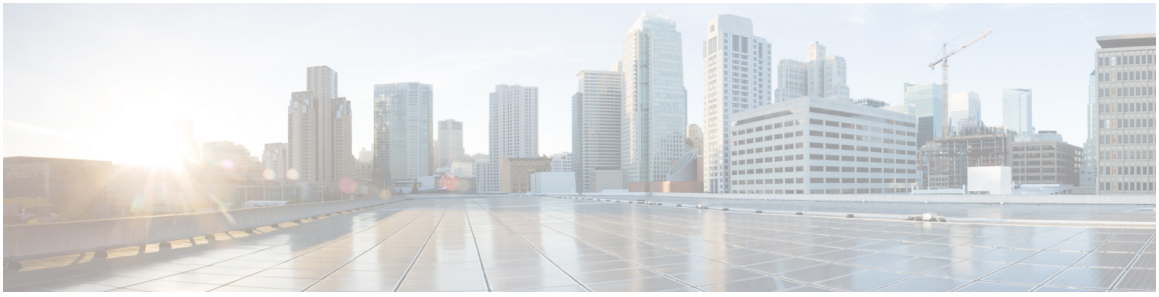
IP Routing Configuration Guide, Cisco IOS XE 17.x

First Published: 2022-11-02

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

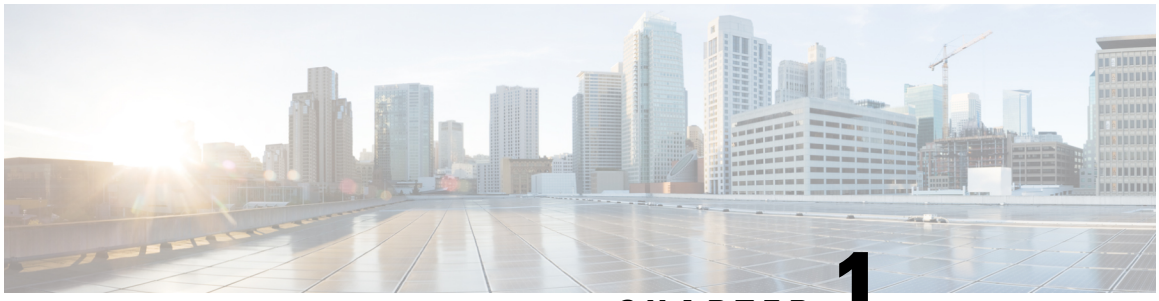
Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

PART I

Protocol Independent



CHAPTER 1

Basic IP Routing

This module describes how to configure basic IP routing. The Internet Protocol (IP) is a network layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network layer protocol in the Internet protocol suite.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Basic IP Routing

Variable-Length Subnet Masks

Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Routing Information Protocol (RIP) Version 2, and static routes support variable-length subnet masks (VLSMs). With VLSMs, you can use different masks for the same network number on different interfaces, which allows you to conserve IP addresses and more efficiently use available address space. However, using VLSMs also presents address assignment challenges for the network administrator and ongoing administrative challenges.

Refer to RFC 1219 for detailed information about VLSMs and how to correctly assign addresses.

Note Consider your decision to use VLSMs carefully. You can easily make mistakes in address assignments and you will generally find that the network is more difficult to monitor using VLSMs.

The best way to implement VLSMs is to keep your existing addressing plan in place and gradually migrate some networks to VLSMs to recover address space.

Static Routes

Static routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the device cannot build a route to a particular destination. They are also useful for specifying a gateway of last resort to which all unroutable packets will be sent.

To configure a static route, use the **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*] global configuration command.

Static routes remains in the device configuration until you remove them (using the **no ip route** global configuration command). However, you can override static routes with dynamic routing information through prudent assignment of administrative distance values. An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is an integer from 0 to 255. In general, the higher the value, the lower the trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored.

Each dynamic routing protocol has a default administrative distance, as listed in the table below. If you want a static route to be overridden by information from a dynamic routing protocol, simply ensure that the administrative distance of the static route is higher than that of the dynamic protocol.

Table 1: Default Administrative Distances for Dynamic Routing Protocols

Route Source	Default Distance
Connected interface	0
Static route	1
Enhanced Interior Gateway Routing Protocol (EIGRP) summary route	5
External Border Gateway Protocol (BGP)	20
Internal EIGRP	90
Interior Gateway Routing Protocol (IGRP)	100
Open Shortest Path First (OSPF)	110
intermediate System to Intermediate System (IS-IS)	115
Routing Information Protocol (RIP)	120
Exterior Gateway Routing Protocol (EGP)	140
On Demand Routing (ODR)	160

Route Source	Default Distance
External EIGRP	170
Internal BGP	200
Unknown	255

Static routes that point to an interface are advertised via RIP, EIGRP, and other dynamic routing protocols, regardless of whether **redistribute static** router configuration commands are specified for those routing protocols. These static routes are advertised because static routes that point to an interface are considered in the routing table to be connected and hence lose their static nature. However, if you define a static route to an interface that is not one of the networks defined in a **network** command, no dynamic routing protocols will advertise the route unless a **redistribute static** command is specified for these protocols.

When an interface goes down, all static routes through that interface are removed from the IP routing table. Also, when the software can no longer find a valid next hop for the address specified as the address of the forwarding device in a static route, the static route is removed from the IP routing table.

Default Routes

Default routes, also known as gateways of last resort, are used to route packets that are addressed to networks not explicitly listed in the routing table. A device might not be able to determine routes to all networks. To provide complete routing capability, network administrators use some devices as smart devices and give the remaining devices default routes to the smart device. (Smart devices have routing table information for the entire internetwork.) Default routes can be either passed along dynamically or configured manually into individual devices.

Most dynamic interior routing protocols include a mechanism for causing a smart device to generate dynamic default information, which is then passed along to other devices.

You can configure a default route by using the following commands:

- **ip default-gateway**
- **ip default-network**
- **ip route 0.0.0.0 0.0.0.0**

You can use the **ip default-gateway** global configuration command to define a default gateway when IP routing is disabled on a device. For instance, if a device is a host, you can use this command to define a default gateway for the device. You can also use this command to transfer a Cisco software image to a device when the device is in boot mode. In boot mode, IP routing is not enabled on the device.

Unlike the **ip default-gateway** command, the **ip default-network** command can be used when IP routing is enabled on a device. When you specify a network by using the **ip default-network** command, the device considers routes to that network for installation as the gateway of last resort on the device.

Gateways of last resort configured by using the **ip default-network** command are propagated differently depending on which routing protocol is propagating the default route. For Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) to propagate the default route, the network specified by the **ip default-network** command must be known to IGRP or EIGRP. The network must be an IGRP- or EIGRP-derived network in the routing table, or the static route used to generate the route to the network must be redistributed into IGRP or EIGRP or advertised into these protocols by using the **network** command. The Routing Information Protocol (RIP) advertises a route to network 0.0.0.0 if a gateway of last

resort is configured by using the **ip default-network** command. The network specified in the **ip default-network** command need not be explicitly advertised under RIP.

Creating a static route to network 0.0.0.0 0.0.0.0 by using the **ip route 0.0.0.0 0.0.0.0** command is another way to set the gateway of last resort on a device. As with the **ip default-network** command, using the static route to 0.0.0.0 is not dependent on any routing protocols. However, IP routing must be enabled on the device. IGRP does not recognize a route to network 0.0.0.0. Therefore, it cannot propagate default routes created by using the **ip route 0.0.0.0 0.0.0.0** command. Use the **ip default-network** command to have IGRP propagate a default route.

EIGRP propagates a route to network 0.0.0.0, but the static route must be redistributed into the routing protocol.

Depending on your release of the Cisco software, the default route created by using the **ip route 0.0.0.0 0.0.0.0** command is automatically advertised by RIP devices. In some releases, RIP does not advertise the default route if the route is not learned via RIP. You might have to redistribute the route into RIP by using the **redistribute** command.

Default routes created using the **ip route 0.0.0.0 0.0.0.0** command are not propagated by Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). Additionally, these default routes cannot be redistributed into OSPF or IS-IS by using the **redistribute** command. Use the **default-information originate** command to generate a default route into an OSPF or IS-IS routing domain.

Default Network

Default networks are used to route packets to destinations not established in the routing table. You can use the **ip default-network** *network-number* global configuration command to configure a default network when IP routing is enabled on the device. When you configure a default network, the device considers routes to that network for installation as the gateway of last resort on the device.

Gateway of Last Resort

When default information is being passed along through a dynamic routing protocol, no further configuration is required. The system periodically scans its routing table to choose the optimal default network as its default route. In the case of the Routing Information Protocol (RIP), there is only one choice, network 0.0.0.0. In the case of Enhanced Interior Gateway Routing Protocol (EIGRP), there might be several networks that can be candidates for the system default. Cisco software uses both administrative distance and metric information to determine the default route (gateway of last resort). The selected default route appears in the gateway of last resort display of the **show ip route** privileged EXEC command.

If dynamic default information is not being passed to the software, candidates for the default route are specified with the **ip default-network** global configuration command. In this usage, the **ip default-network** command takes an unconnected network as an argument. If this network appears in the routing table from any source (dynamic or static), it is flagged as a candidate default route and is a possible choice as the default route.

If the device has no interface on the default network, but does have a route to it, it considers this network as a candidate default path. The route candidates are examined and the best one is chosen, based on administrative distance and metric. The gateway to the best default path becomes the gateway of last resort.

Maximum Number of Paths

By default, most IP routing protocols install a maximum of four parallel routes in a routing table. Static routes always install six routes. The exception is Border Gateway Protocol (BGP), which by default allows only one path (the best path) to a destination. However, BGP can be configured to use equal and unequal cost multipath load sharing.

The number of parallel routes that you can configure to be installed in the routing table is dependent on the installed version of Cisco software. To change the maximum number of parallel paths allowed, use the **maximum-paths** *number-paths* command in router configuration mode.

Multi-Interface Load Splitting

Multi-interface load splitting allows you to efficiently control traffic that travels across multiple interfaces to the same destination. The **traffic-share min** router configuration command specifies that if multiple paths are available to the same destination, only paths with the minimum metric will be installed in the routing table. The number of paths allowed is never more than six. For dynamic routing protocols, the number of paths is controlled by the **maximum-paths** router configuration command. The static route source can install six paths. If more paths are available, the extra paths are discarded. If some installed paths are removed from the routing table, pending routes are added automatically.

Routing Information Redistribution

In addition to running multiple routing protocols simultaneously, Cisco software can be configured to redistribute information from one routing protocol to another. For example, you can configure a device to readvertise Enhanced Interior Gateway Routing Protocol (EIGRP)-derived routes using the Routing Information Protocol (RIP), or to readvertise static routes using the EIGRP protocol. Redistribution from one routing protocol to another can be configured in all of the IP-based routing protocols.

You also can conditionally control the redistribution of routes between routing domains by configuring route maps between the two domains. A route map is a route/packet filter that is configured with permit and deny statements, match and set clauses, and sequence numbers.

Although redistribution is a protocol-independent feature, some of the **match** and **set** commands are specific to a particular protocol.

One or more **match** commands and one or more **set** commands are configured in route map configuration mode. If there are no **match** commands, then everything matches. If there are no **set** commands, then no set action is performed.

To define a route map for redistribution, use the **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] global configuration command.

The metrics of one routing protocol do not necessarily translate into the metrics of another. For example, the RIP metric is a hop count and the EIGRP metric is a combination of five metric values. In such situations, a dynamic metric is assigned to the redistributed route. Redistribution in these cases should be applied consistently and carefully with inbound filtering to avoid routing loops.

Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting.

Supported Metric Translations

This section describes supported automatic metric translations between the routing protocols. The following descriptions assume that you have not defined a default redistribution metric that replaces metric conversions:

- The Routing Information Protocol (RIP) can automatically redistribute static routes. It assigns static routes a metric of 1 (directly connected).
- The Border Gateway Protocol (BGP) does not normally send metrics in its routing updates.

- The Enhanced Interior Gateway Routing Protocol (EIGRP) can automatically redistribute static routes from other EIGRP-routed autonomous systems as long as the static route and any associated interfaces are covered by an EIGRP network statement. EIGRP assigns static routes a metric that identifies them as directly connected. EIGRP does not change the metrics of routes derived from EIGRP updates from other autonomous systems.

Note Note that any protocol can redistribute routes from other routing protocols as long as a default metric is configured.

Protocol Differences in Implementing the no redistribute Command

Caution Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting. In most cases, changing or disabling any keyword will not affect the state of other keywords.

Different protocols implement the **no redistribute** command differently as follows:

- In Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP) configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.
- The **no redistribute isis** command removes the Intermediate System to Intermediate System (IS-IS) redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- The Enhanced Interior Gateway Routing Protocol (EIGRP) used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.

Sources of Routing Information Filtering

Filtering sources of routing information prioritizes routing information from different sources because some pieces of routing information might be more accurate than others. An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual device or a group of devices. In a large network, some routing protocols and some devices can be more reliable than others as sources of routing information. Also, when multiple routing processes are running in the same device for IP, the same route could be advertised by more than one routing process. By specifying administrative distance values, you enable the device to intelligently discriminate between sources of routing information. The device always picks the route whose routing protocol has the lowest administrative distance.

There are no general guidelines for assigning administrative distances because each network has its own requirements. You must determine a reasonable matrix of administrative distances for the network as a whole.

For example, consider a device using the Enhanced Interior Gateway Routing Protocol (EIGRP) and the Routing Information Protocol (RIP). Suppose you trust the EIGRP-derived routing information more than the

RIP-derived routing information. In this example, because the default EIGRP administrative distance is lower than the default RIP administrative distance, the device uses the EIGRP-derived information and ignores the RIP-derived information. However, if you lose the source of the EIGRP-derived information (because of a power shutdown at the source network, for example), the device uses the RIP-derived information until the EIGRP-derived information reappears.

Note You can also use administrative distance to rate the routing information from devices that are running the same routing protocol. This application is generally discouraged if you are unfamiliar with this particular use of administrative distance because it can result in inconsistent routing information, including forwarding loops.

Note The weight of a route can no longer be set with the **distance** command. To set the weight for a route, use a route map.

Authentication Key Management and Supported Protocols

Key management is a method of controlling the authentication keys used by routing protocols. Not all protocols support key management. Authentication keys are available for Director Response Protocol (DRP) Agent, Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP) Version 2.

You can manage authentication keys by defining key chains, identifying the keys that belong to the key chain, and specifying how long each key is valid. Each key has its own key identifier (specified using the **key chain** configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and the message digest algorithm 5 (MD5) authentication key in use.

You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes.

How to Configure Basic IP Routing

Redistributing Routing Information

You can redistribute routes from one routing domain into another, with or without controlling the redistribution with a route map. To control which routes are redistributed, configure a route map and reference the route map from the **redistribute** command.

The tasks in this section describe how to define the conditions for redistributing routes (a route map), how to redistribute routes, and how to remove options for redistributing routes, depending on the protocol being used.

Defining Conditions for Redistributing Routes

Route maps can be used to control route redistribution (or to implement policy-based routing). To define conditions for redistributing routes from one routing protocol into another, configure the **route-map** command.

Then use at least one **match** command in route map configuration mode, as needed. At least one **match** command is used in this task because the purpose of the task is to illustrate how to define one or more conditions on which to base redistribution.

Note A route map is not required to have **match** commands; it can have only **set** commands. If there are no **match** commands, everything matches the route map.

Note There are many more **match** commands not shown in this table. For additional **match** commands, see the *Cisco IOS Master Command List*.

Command or Action	Purpose
match as-path *path-list-number*	Matches a BGP autonomous system path access list.
match community {*standard-list-number* | *expanded-list-number* | *community-list-name* **match community**[**exact**]}	Matches a BGP community.
match ip address {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}	Matches routes that have a destination network address that is permitted to policy route packets or is permitted by a standard access list, an extended access list, or a prefix list.
match metric *metric-value*	Matches routes with the specified metric.
match ip next-hop {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches a next-hop device address passed by one of the specified access lists.
match tag *tag-value* [*tag-value*]	Matches the specified tag value.
match interface *type number* [*type number*]	Matches routes that use the specified interface as the next hop.
match ip route-source {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches the address specified by the advertised access lists.
match route-type {**local** | **internal** | **external** [**type-1** | **type-2**] | **level-1** | **level-2**}	Matches the specified route type.

To optionally specify the routing actions for the system to perform if the match criteria are met (for routes that are being redistributed by the route map), use one or more **set** commands in route map configuration mode, as needed.

Note A route map is not required to have **set** commands; it can have only **match** commands.

Note There are more **set** commands not shown in this table. For additional **set** commands, see the *Cisco IOS Master Command List*.

Command or Action	Purpose
set community {*community-number* [**additive**] [**well-known**]| **none**}	Sets the community attribute (for BGP).
set dampening *halflife reuse suppress max-suppress-time*	Sets route dampening parameters (for BGP).
set local-preference *number-value*	Assigns a local preference value to a path (for BGP).
set origin {**igp** | **egp** *as-number* | **incomplete**}	Sets the route origin code.
set as-path{**tag** | **prepend** *as-path-string* }	Modifies the autonomous system path (for BGP).
set next-hop *next-hop*	Specifies the address of the next hop.
set automatic-tag	Enables automatic computation of the tag table.
set level {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**}	Specifies the areas to import routes.
set metric *metric-value*	Sets the metric value for redistributed routes (for any protocol, except EIGRP).
set metric *bandwidth delay reliability load mtu*	Sets the metric value for redistributed routes (for EIGRP only).
set metric-type {**internal** | **external** | **type-1** | **type-2**}	Sets the metric type for redistributed routes.

Command or Action	Purpose
set metric-type internal	Sets the Multi Exit Discriminator (MED) value on prefixes advertised to the external BGP neighbor to match the Interior Gateway Protocol (IGP) metric of the next hop.
set tag *tag-value*	Sets a tag value to be applied to redistributed routes.

Redistributing Routes from One Routing Domain to Another

Perform this task to redistribute routes from one routing domain into another and to control route redistribution. This task shows how to redistribute OSPF routes into a BGP domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system*
4. **redistribute** *protocol process-id*
5. **default-metric** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 109`	Enables a BGP routing process and enters router configuration mode.
Step 4	**redistribute** *protocol process-id* **Example:** `Device(config-router)# redistribute ospf 2 1`	Redistributes routes from the specified routing domain into another routing domain.
Step 5	**default-metric** *number* **Example:**	Sets the default metric value for redistributed routes.

	Command or Action	Purpose
	`Device(config-router)# default-metric 10`	**Note** The metric value specified in the **redistribute** command supersedes the metric value specified using the **default-metric** command.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Removing Options for Redistribution Routes

Caution Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting.

Different protocols implement the **no redistribute** command differently as follows:

- In BGP, OSPF, and RIP configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.
- The **no redistribute isis** command removes the IS-IS redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- EIGRP used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.
- For the **no redistribute connected** command, the behavior is subtractive if the **redistribute** command is configured under the **router bgp** or the **router ospf** command. The behavior is complete removal of the command if it is configured under the **router isis** or the **router eigrp** command.

The following OSPF commands illustrate how various options are removed from the redistribution in router configuration mode.

Command or Action	Purpose
`no redistribute connected metric 1000 subnets`	Removes the configured metric value of 1000 and the configured subnets and retains the **redistribute connected** command in the configuration.

Command or Action	Purpose
`no redistribute connected metric 1000`	Removes the configured metric value of 1000 and retains the **redistribute connected subnets** command in the configuration.
`no redistribute connected subnets`	Removes the configured subnets and retains the **redistribute connected metric** *metric-value* command in the configuration.
`no redistribute connected`	Removes the **redistribute connected** command and any of the options that were configured for the command.

Configuring Routing Information Filtering

Note When routes are redistributed between Open Shortest Path First (OSPF) processes, no OSPF metrics are preserved.

Controlling the Advertising of Routes in Routing Updates

To prevent other devices from learning one or more routes, you can suppress routes from being advertised in routing updates. To suppress routes from being advertised in routing updates, use the **distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing-process* | *as-number*] command in router configuration mode.

You cannot specify an interface name in Open Shortest Path First (OSPF). When used for OSPF, this feature applies only to external routes.

Controlling the Processing of Routing Updates

You might want to avoid processing certain routes that are listed in incoming updates (this does not apply to Open Shortest Path First [OSPF] or Intermediate System to Intermediate System [IS-IS]). To suppress routes in incoming updates, use the **distribute-list** {*access-list-number* | *access-list-name*} **in** [*interface-type interface-number*] command in router configuration mode.

Filtering Sources of Routing Information

To filter sources of routing information, use the **distance** *ip-address wildcard- mask* [*ip-standard-acl* | *ip-extended-acl* | *access-list-name*] command in router configuration mode.

Managing Authentication Keys

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *number*
5. **key-string** *text*
6. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
7. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
8. **end**
9. **show key chain**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes. `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain chain1`	Defines a key chain and enters key-chain configuration mode.
Step 4	**key** *number* **Example:** `Device(config-keychain)# key 1`	Identifies number of an authentication key on a key chain. The range of keys is from 0 to 2147483647. The key identification numbers need not be consecutive.
Step 5	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string string1`	Identifies the key string.

	Command or Action	Purpose
Step 6	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# accept-lifetime 13:30:00 Dec 22 2011 duration 7200`	Specifies the time period during which the key can be received.
Step 7	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime 14:30:00 Dec 22 2011 duration 3600`	Specifies the time period during which the key can be sent.
Step 8	**end** **Example:** `Device(config-keychain-key)# end`	Exits key-chain key configuration mode and returns to privileged EXEC mode.
Step 9	**show key chain** **Example:** `Device# show key chain`	(Optional) Displays authentication key information.

Monitoring and Maintaining the IP Network

Clearing Routes from the IP Routing Table

You can remove all contents of a particular table. Clearing a table may become necessary when the contents of the particular structure have become, or are suspected to be, invalid.

To clear one or more routes from the IP routing table, use the **clear ip route** {*network* [*mask*] | *****} command in privileged EXEC mode.

Displaying System and Network Statistics

You can use the following **show** commands to display system and network statistics. You can display specific statistics such as contents of IP routing tables, caches, and databases. You can also display information about node reachability and discover the routing path that packets leaving your device are taking through the network. This information can an be used to determine resource utilization and solve network problems.

Command or Action	Purpose
show ip cache policy	Displays cache entries in the policy route cache.
show ip local policy	Displays the local policy route map if one exists.
show ip policy	Displays policy route maps.

Command or Action	Purpose
show ip protocols	Displays the parameters and current state of the active routing protocols.
show ip route [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **list** {**access-list-number** | *access-list-name*} | **static download**]	Displays the current state of the routing table.
show ip route summary	Displays the current state of the routing table in summary form.
show ip route supernets-only	Displays supernets.
show key chain [*name-of-chain*]	Displays authentication key information.
show route-map [*map-name*]	Displays all route maps configured or only the one specified.

Configuration Examples for Basic IP Routing

Example: Variable-Length Subnet Mask

The following example uses two different subnet masks for the class B network address of 172.16.0.0. A subnet mask of /24 is used for LAN interfaces. The /24 mask allows 265 subnets with 254 host IP addresses on each subnet. The final subnet of the range of possible subnets using a /24 mask (172.16.255.0) is reserved for use on point-to-point interfaces and assigned a longer mask of /30. The use of a /30 mask on 172.16.255.0 creates 64 subnets (172.16.255.0 to 172.16.255.252) with 2 host addresses on each subnet.

Caution: To ensure unambiguous routing, you must not assign 172.16.255.0/24 to a LAN interface in your network.

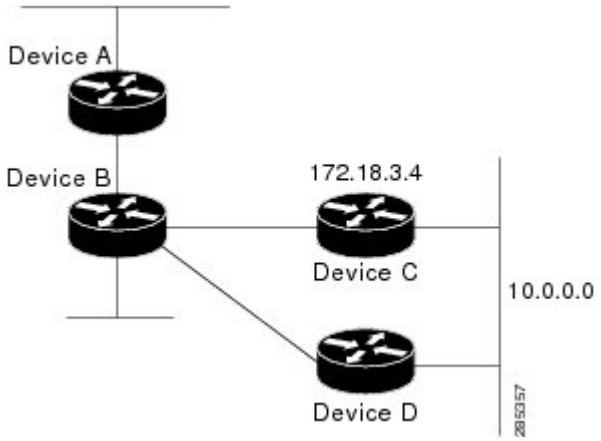
```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 172.16.1.1 255.255.255.0
Device(config-if)# ! 8 bits of host address space reserved for GigabitEthernet interfaces
Device(config-if)# exit
Device(config)# interface Serial 0/0/0
Device(config-if)# ip address 172.16.255.5 255.255.255.252
Device(config-if)# ! 2 bits of address space reserved for point-to-point serial interfaces
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 172.16.0.0
Device(config-router)# ! Specifies the network directly connected to the device
```

Example: Overriding Static Routes with Dynamic Protocols

In the following example, packets for network 10.0.0.0 from Device B (where the static route is installed) will be routed through 172.18.3.4 if a route with an administrative distance less than 110 is not available. The figure below illustrates this example. The route learned by a protocol with an administrative distance of less than 110 might cause Device B to send traffic destined for network 10.0.0.0 via the alternate path through Device D.

```
Device(config)# ip route 10.0.0.0 255.0.0.0 172.18.3.4 110
```

Figure 1: Overriding Static Routes



Example: IP Default Gateway as a Static IP Next Hop When IP Routing Is Disabled

The following example shows how to configure IP address 172.16.5.4 as the default route when IP routing is disabled:

```
Device> enable
Device# configure terminal
Device(conf)# no ip routing
Device(conf)# ip default-gateway 172.16.15.4
```

Examples: Administrative Distances

In the following example, the **router eigrp** global configuration command configures Enhanced Interior Gateway Routing Protocol (EIGRP) routing in autonomous system 1. The **network** command configuration specifies EIGRP routing on networks 192.168.7.0 and 172.16.0.0. The first **distance** router configuration command sets the default administrative distance to 255, which instructs the device to ignore all routing updates from devices for which an explicit distance has not been set. The second **distance** command sets the administrative distance to 80 for internal EIGRP routes and to 100 for external EIGRP routes. The third **distance** command sets the administrative distance to 120 for the device with the address 172.16.1.3.

```
Device(config)# router eigrp 1
Device(config-router)# network 192.168.7.0
```

```
Device(config-router)# network 172.16.0.0
Device(config-router)# distance 255
Device(config-router)# distance eigrp 80 100
Device(config-router)# distance 120 172.16.1.3 0.0.0.0
```

Note The **distance eigrp** command must be used to set the administrative distance for EIGRP-derived routes.

The following example assigns the device with the address 192.168.7.18 an administrative distance of 100 and all other devices on subnet 192.168.7.0 an administrative distance of 200:

```
Device(config-router)# distance 100 192.168.7.18 0.0.0.0
Device(config-router)# distance 200 192.168.7.0 0.0.0.255
```

However, if you reverse the order of these two commands, all devices on subnet 192.168.7.0 are assigned an administrative distance of 200, including the device at address 192.168.7.18:

```
Device(config-router)# distance 200 192.168.7.0 0.0.0.255
Device(config-router)# distance 100 192.168.7.18 0.0.0.0
```

Note Assigning administrative distances can be used to solve unique problems. However, administrative distances should be applied carefully and consistently to avoid the creation of routing loops or other network failures.

In the following example, the distance value for IP routes learned is 90. Preference is given to these IP routes rather than routes with the default administrative distance value of 110.

```
Device(config)# router isis
Device(config-router)# distance 90 ip
```

Example: Static Routing Redistribution

In the example that follows, three static routes are specified, two of which are to be advertised. The static routes are created by specifying the **redistribute static** router configuration command and then specifying an access list that allows only those two networks to be passed to the Enhanced Interior Gateway Routing Protocol (EIGRP) process. Any redistributed static routes should be sourced by a single device to minimize the likelihood of creating a routing loop.

```
Device(config)# ip route 192.168.2.0 255.255.255.0 192.168.7.65
Device(config)# ip route 192.168.5.0 255.255.255.0 192.168.7.65
Device(config)# ip route 10.10.10.0 255.255.255.0 10.20.1.2
Device(config)# !
Device(config)# access-list 3 permit 192.168.2.0 0.0.255.255
Device(config)# access-list 3 permit 192.168.5.0 0.0.255.255
Device(config)# access-list 3 permit 10.10.10.0 0.0.0.255
Device(config)# !
Device(config)# router eigrp 1
Device(config-router)# network 192.168.0.0
Device(config-router)# network 10.10.10.0
Device(config-router)# redistribute static metric 10000 100 255 1 1500
Device(config-router)# distribute-list 3 out static
```

Examples: EIGRP Redistribution

Each Enhanced Interior Gateway Routing Protocol (EIGRP) routing process provides routing information to only one autonomous system. The Cisco software must run a separate EIGRP process and maintain a separate routing database for each autonomous system that it services. However, you can transfer routing information between these routing databases.

In the following configuration, network 10.0.0.0 is configured under EIGRP autonomous system 1 and network 192.168.7.0 is configured under EIGRP autonomous system 101:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# exit
Device(config)# router eigrp 101
Device(config-router)# network 192.168.7.0
```

In the following example, routes from the 192.168.7.0 network are redistributed into autonomous system 1 (without passing any other routing information from autonomous system 101):

```
Device(config)# access-list 3 permit 192.168.7.0
Device(config)# !
Device(config)# route-map 101-to-1 permit 10
Device(config-route-map)# match ip address 3
Device(config-route-map)# set metric 10000 100 1 255 1500
Device(config-route-map)# exit
Device(config)# router eigrp 1
Device(config-router)# redistribute eigrp 101 route-map 101-to-1
Device(config-router)#!
```

The following example is an alternative way to redistribute routes from the 192.168.7.0 network into autonomous system 1. Unlike the previous configuration, this method does not allow you to set the metric for redistributed routes.

```
Device(config)# access-list 3 permit 192.168.7.0
Device(config)# !
Device(config)# router eigrp 1
Device(config-router)# redistribute eigrp 101
Device(config-router)# distribute-list 3 out eigrp 101
Device(config-router)# !
```

Example: Mutual Redistribution Between EIGRP and RIP

Consider a WAN at a university that uses the Routing Information Protocol (RIP) as an interior routing protocol. Assume that the university wants to connect its WAN to regional network 172.16.0.0, which uses the Enhanced Interior Gateway Routing Protocol (EIGRP) as the routing protocol. The goal in this case is to advertise the networks in the university network to devices in the regional network.

Mutual redistribution is configured between EIGRP and RIP in the following example:

```
Device(config)# access-list 10 permit 172.16.0.0
Device(config)# !
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# redistribute rip metric 10000 100 255 1 1500
Device(config-router)# default-metric 10
Device(config-router)# distribute-list 10 out rip
```

```
Device(config-router)# exit
Device(config)# router rip
Device(config-router)# redistribute eigrp 1
Device(config-router)# !
```

In this example, an EIGRP routing process is started. The **network** router configuration command specifies that network 172.16.0.0 (the regional network) is to send and receive EIGRP routing information. The **redistribute** router configuration command specifies that RIP-derived routing information be advertised in routing updates. The **default-metric** router configuration command assigns an EIGRP metric to all RIP-derived routes. The **distribute-list** router configuration command instructs the Cisco software to use access list 10 (not defined in this example) to limit the entries in each outgoing update. The access list prevents unauthorized advertising of university routes to the regional network.

Example: Mutual Redistribution Between EIGRP and BGP

In the following example, mutual redistribution is configured between the Enhanced Interior Gateway Routing Protocol (EIGRP) and the Border Gateway Protocol (BGP).

Routes from EIGRP routing process 101 are injected into BGP autonomous system 50000. A filter is configured to ensure that the correct routes are advertised, in this case, three networks. Routes from BGP autonomous system 50000 are injected into EIGRP routing process 101. The same filter is used.

```
Device(config)# ! All networks that should be advertised from R1 are controlled with ACLs:

Device(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.16.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.25.0.0 0.0.255.255
Device(config)# ! Configuration for router R1:
Device(config)# router bgp 50000
Device(config-router)# network 172.18.0.0
Device(config-router)# network 172.16.0.0
Device(config-router)# neighbor 192.168.10.1 remote-as 2
Device(config-router)# neighbor 192.168.10.15 remote-as 1
Device(config-router)# neighbor 192.168.10.24 remote-as 3
Device(config-router)# redistribute eigrp 101
Device(config-router)# distribute-list 1 out eigrp 101
Device(config-router)# exit
Device(config)# router eigrp 101
Device(config-router)# network 172.25.0.0
Device(config-router)# redistribute bgp 50000
Device(config-router)# distribute-list 1 out bgp 50000
Device(config-router)# !
```

Caution BGP should be redistributed into an Interior Gateway Protocol (IGP) when there are no other suitable options. Redistribution from BGP into any IGP should be applied with proper filtering by using distribute lists, IP prefix lists, and route map statements to limit the number of prefixes.

Examples: OSPF Routing and Route Redistribution

OSPF typically requires coordination among many internal devices, area border routers (ABRs), and Autonomous System Boundary Routers (ASBRs). At a minimum, OSPF-based devices can be configured with all default parameter values, with no authentication, and with interfaces assigned to areas.

This section provides the following configuration examples:

- The first example shows simple configurations illustrating basic OSPF commands.
- The second example shows configurations for an internal device, ABR, and ASBR within a single, arbitrarily assigned OSPF autonomous system.
- The third example illustrates a more complex configuration and the application of various tools available for controlling OSPF-based routing environments.

Examples: Basic OSPF Configuration

The following example illustrates a simple OSPF configuration that enables OSPF routing process 1, attaches Gigabit Ethernet interface 0/0/0 to area 0.0.0.0, and redistributes RIP into OSPF and OSPF into RIP:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 172.16.1.1 255.255.255.0
Device(config-if)# ip ospf cost 1
Device(config-if)# exit
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# ip address 172.17.1.1 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# network 172.18.0.0 0.0.255.255 area 0.0.0.0
Device(config-router)# redistribute rip metric 1 subnets
Device(config-router)# exit
Device(config)# router rip
Device(config-router)# network 172.17.0.0
Device(config-router)# redistribute ospf 1
Device(config-router)# default-metric 1
Device(config-router)# !
```

The following example illustrates the assignment of four area IDs to four IP address ranges. In the example, OSPF routing process 1 is initialized, and four OSPF areas are defined: 10.9.50.0, 2, 3, and 0. Areas 10.9.50.0, 2, and 3 mask specific address ranges, whereas area 0 enables OSPF for all other networks.

```
Device(config)# router ospf 1
Device(config-router)# network 172.18.20.0 0.0.0.255 area 10.9.50.0
Device(config-router)# network 172.18.0.0 0.0.255.255 area 2
Device(config-router)# network 172.19.10.0 0.0.0.255 area 3
Device(config-router)# network 0.0.0.0 255.255.255.255 area 0
Device(config-router)# exit
Device(config)# ! GigabitEthernet interface 0/0/0 is in area 10.9.50.0:
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 172.18.20.5 255.255.255.0
Device(config-if)# exit
Device(config)# ! GigabitEthernet interface 1/0/0 is in area 2:
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# ip address 172.18.1.5 255.255.255.0
Device(config-if)# exit
Device(config)# ! GigabitEthernet interface 2/0/0 is in area 2:
Device(config)# interface GigabitEthernet 2/0/0
Device(config-if)# ip address 172.18.2.5 255.255.255.0
Device(config-if)# exit
Device(config)# ! GigabitEthernet interface 3/0/0 is in area 3:
Device(config)# interface GigabitEthernet 3/0/0
Device(config-if)# ip address 172.19.10.5 255.255.255.0
Device(config-if)# exit
Device(config)# ! GigabitEthernet interface 4/0/0 is in area 0:
```

```
Device(config)# interface GigabitEthernet 4/0/0
Device(config-if)# ip address 172.19.1.1 255.255.255.0
Device(config-if)# exit
Device(config)# ! GigabitEthernet interface 5/0/0 is in area 0:
Device(config)# interface GigabitEthernet 5/0/0
Device(config-if)# ip address 10.1.0.1 255.255.0.0
Device(config-if)# !
```

Each **network** router configuration command is evaluated sequentially, so the specific order of these commands in the configuration is important. The Cisco software sequentially evaluates the *address/wildcard-mask* pair for each interface. See the *IP Routing Protocols Command Reference* for more information.

Consider the first **network** command. Area ID 10.9.50.0 is configured for the interface on which subnet 172.18.20.0 is located. Assume that a match is determined for Gigabit Ethernet interface 0/0/0. Gigabit Ethernet interface 0/0/0 is attached to Area 10.9.50.0 only.

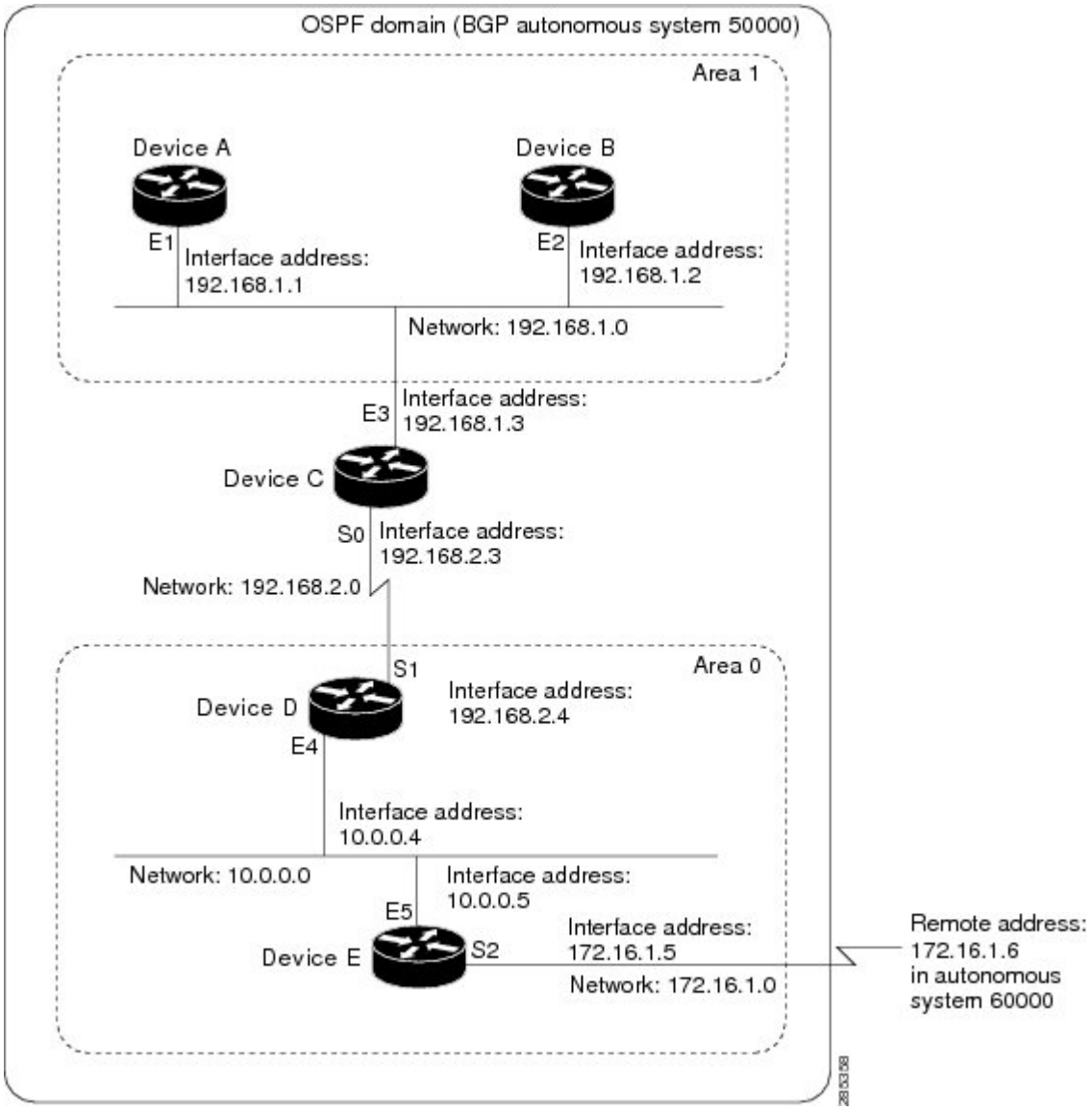
The second **network** command is evaluated next. For Area 2, the same process is then applied to all interfaces (except Gigabit Ethernet interface 0/0/0). Assume that a match is determined for Gigabit Ethernet interface 1/0/0. OSPF is then enabled for that interface, and Gigabit Ethernet 1/0/0 is attached to Area 2.

This process of attaching interfaces to OSPF areas continues for all **network** commands. Note that the last **network** command in this example is a special case. With this command, all available interfaces (not explicitly attached to another area) are attached to Area 0.

Example: Internal Device ABR and ASBRs Configuration

The figure below provides a general network map that illustrates a sample configuration for several devices within a single OSPF autonomous system.

Figure 2: Example OSPF Autonomous System Network Map



In this configuration, five devices are configured in OSPF autonomous system 1:

- Device A and Device B are both internal devices within area 1.
- Device C is an OSPF ABR. Note that for Device C, area 1 is assigned to E3 and Area 0 is assigned to S0.
- Device D is an internal device in area 0 (backbone area). In this case, both **network** router configuration commands specify the same area (area 0, or the backbone area).
- Device E is an OSPF ASBR. Note that the Border Gateway Protocol (BGP) routes are redistributed into OSPF and that these routes are advertised by OSPF.

Note Definitions of all areas in an OSPF autonomous system need not be included in the configuration of all devices in the autonomous system. You must define only the *directly* connected areas. In the example that follows, routes in Area 0 are learned by the devices in area 1 (Device A and Device B) when the ABR (Device C) injects summary link state advertisements (LSAs) into area 1.

Autonomous system 60000 is connected to the outside world via the BGP link to the external peer at IP address 172.16.1.6.

Following is the sample configuration for the general network map shown in the figure above.

Device A Configuration--Internal Device

```
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# ip address 192.168.1.1 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# network 192.168.1.0 0.0.0.255 area 1
Device(config-router)# exit
```

Device B Configuration--Internal Device

```
Device(config)# interface GigabitEthernet 2/0/0
Device(config-if)# ip address 192.168.1.2 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# network 192.168.1.0 0.0.0.255 area 1
Device(config-router)# exit
```

Device C Configuration--ABR

```
Device(config)# interface GigabitEthernet 3/0/0
Device(config-if)# ip address 192.168.1.3 255.255.255.0
Device(config-if)# exit
Device(config)# interface Serial 0/0/0
Device(config-if)# ip address 192.168.2.3 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# network 192.168.1.0 0.0.0.255 area 1
Device(config-router)# network 192.168.2.0 0.0.0.255 area 0
Device(config-router)# exit
```

Device D Configuration--Internal Device

```
Device(config)# interface GigabitEthernet 4/0/0
Device(config-if)# ip address 10.0.0.4 255.0.0.0
Device(config-if)# exit
Device(config)# interface Serial 1/0/0
Device(config-if)# ip address 192.168.2.4 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# network 192.168.2.0 0.0.0.255 area 0
Device(config-router)# network 10.0.0.0 0.255.255.255 area 0
Device(config-router)# exit
```

Device E Configuration--ASBR

```
Device(config)# interface GigabitEthernet 5/0/0
Device(config-if)# ip address 10.0.0.5 255.0.0.0
Device(config-if)# exit
Device(config)# interface Serial 2/0/0
Device(config-if)# ip address 172.16.1.5 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# network 10.0.0.0 0.255.255.255 area 0
Device(config-router)# redistribute bgp 50000 metric 1 metric-type 1
Device(config-router)# exit
Device(config)# router bgp 50000
Device(config-router)# network 192.168.0.0
Device(config-router)# network 10.0.0.0
Device(config-router)# neighbor 172.16.1.6 remote-as 60000
```

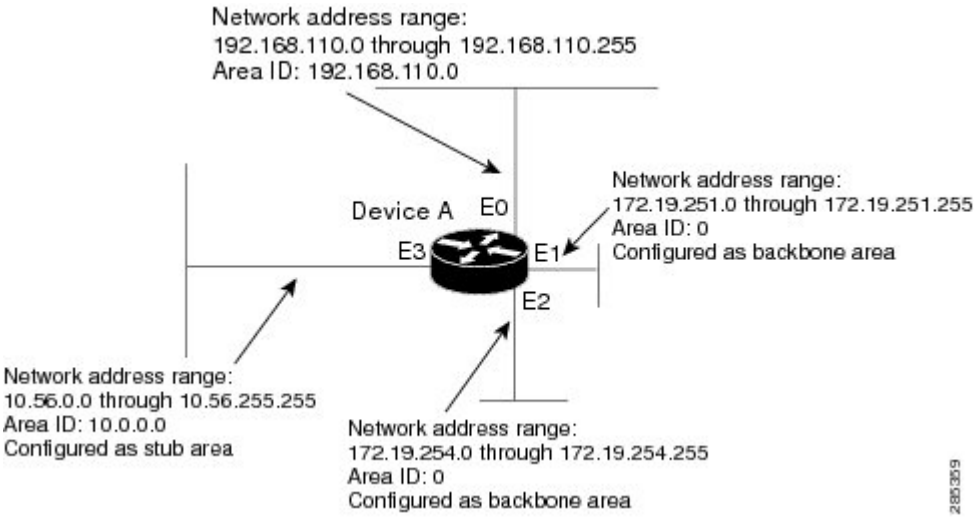
Example: Complex OSPF Configuration

The following sample configuration accomplishes several tasks in setting up an ABR. These tasks can be split into two general categories:

- Basic OSPF configuration
- Route redistribution

The specific tasks outlined in this configuration are detailed briefly in the following descriptions. The figure below illustrates the network address ranges and area assignments for the interfaces.

Figure 3: Interface and Area Specifications for OSPF Configuration Example



The basic configuration tasks in this example are as follows:

- Configure address ranges for Gigabit Ethernet interface 0/0/0 through Gigabit Ethernet interface 3/0/0.
- Enable OSPF on each interface.
- Set up an OSPF authentication password for each area and network.
- Assign link-state metrics and other OSPF interface configuration options.

- Create a stub area with area ID 10.0.0.0. (Note that the **authentication** and **stub** options of the **area** router configuration command are specified with separate **area** command entries, but they can be merged into a single **area** command.)
- Specify the backbone area (area 0).

Configuration tasks associated with redistribution are as follows:

- Redistribute the Enhanced Interior Gateway Routing Protocol (EIGRP) and the Routing Information Protocol (RIP) into OSPF with various options set (including metric-type, metric, tag, and subnet).
- Redistribute EIGRP and OSPF into RIP.

The following is an example OSPF configuration:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 192.168.110.201 255.255.255.0
Device(config-if)# ip ospf authentication-key abcdefgh
Device(config-if)# ip ospf cost 10
Device(config-if)# exit
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# ip address 172.19.251.201 255.255.255.0
Device(config-if)# ip ospf authentication-key ijklmnop
Device(config-if)# ip ospf cost 20
Device(config-if)# ip ospf retransmit-interval 10
Device(config-if)# ip ospf transmit-delay 2
Device(config-if)# ip ospf priority 4
Device(config-if)# exit
Device(config)# interface GigabitEthernet 2/0/0
Device(config-if)# ip address 172.19.254.201 255.255.255.0
Device(config-if)# ip ospf authentication-key abcdefgh
Device(config-if)# ip ospf cost 10
Device(config-if)# exit
Device(config)# interface GigabitEthernet 3/0/0
Device(config-if)# ip address 10.56.0.201 255.255.0.0
Device(config-if)# ip ospf authentication-key ijklmnop
Device(config-if)# ip ospf cost 20
Device(config-if)# ip ospf dead-interval 80
Device(config-if)# exit
```

In the following configuration, OSPF is on network 172.19.0.0:

```
Device(config)# router ospf 1
Device(config-router)# network 10.0.0.0 0.255.255.255 area 10.0.0.0
Device(config-router)# network 192.168.110.0 0.0.0.255 area 192.168.110.0
Device(config-router)# network 172.19.0.0 0.0.255.255 area 0
Device(config-router)# area 0 authentication
Device(config-router)# area 10.0.0.0 stub
Device(config-router)# area 10.0.0.0 authentication
Device(config-router)# area 10.0.0.0 default-cost 20
Device(config-router)# area 192.168.110.0 authentication
Device(config-router)# area 10.0.0.0 range 10.0.0.0 255.0.0.0
Device(config-router)# area 192.168.110.0 range 192.168.110.0 255.255.255.0
Device(config-router)# area 0 range 172.19.251.0 255.255.255.0
Device(config-router)# area 0 range 172.19.254.0 255.255.255.0
Device(config-router)# redistribute eigrp 200 metric-type 2 metric 1 tag 200 subnets
Device(config-router)# redistribute rip metric-type 2 metric 1 tag 200
Device(config-router)# exit
```

In the following configuration, EIGRP autonomous system 1 is on 172.19.0.0:

```
Device(config)# router eigrp 1
Device(config-router)# network 172.19.0.0
Device(config-router)# exit
Device(config)# ! RIP for 192.168.110.0:
Device(config)# router rip
Device(config-router)# network 192.168.110.0
Device(config-router)# redistribute eigrp 1 metric 1
Device(config-router)# redistribute ospf 201 metric 1
Device(config-router)# exit
```

Example: Default Metric Values Redistribution

The following example shows a device in autonomous system 1 that is configured to run both the Routing Information Protocol (RIP) and the Enhanced Interior Gateway Routing Protocol (EIGRP). The example advertises EIGRP-derived routes using RIP and assigns the EIGRP-derived routes a RIP metric of 10.

```
Device(config)# router rip
Device(config-router)# redistribute eigrp 1
Device(config-router)# default-metric 10
Device(config-router)# exit
```

Examples: Redistribution With and Without Route Maps

The examples in this section illustrate the use of redistribution, with and without route maps. Examples from both the IP and Connectionless Network Service (CLNS) routing protocols are given. The following example redistributes all Open Shortest Path First (OSPF) routes into the Enhanced Interior Gateway Routing Protocol (EIGRP):

```
Device(config)# router eigrp 1
Device(config-router)# redistribute ospf 101
Device(config-router)# exit
```

The following example redistributes Routing Information Protocol (RIP) routes with a hop count equal to 1 into OSPF. These routes will be redistributed into OSPF as external link state advertisements (LSAs) with a metric of 5, metric a type of type 1, and a tag equal to 1.

```
Device(config)# router ospf 1
Device(config-router)# redistribute rip route-map rip-to-ospf
Device(config-router)# exit
Device(config)# route-map rip-to-ospf permit
Device(config-route-map)# match metric 1
Device(config-route-map)# set metric 5
Device(config-route-map)# set metric-type type 1
Device(config-route-map)# set tag 1
Device(config-route-map)# exit
```

The following example redistributes OSPF learned routes with tag 7 as a RIP metric of 15:

```
Device(config)# router rip
Device(config-router)# redistribute ospf 1 route-map 5
Device(config-router)# exit
Device(config)# route-map 5 permit
Device(config-route-map)# match tag 7
Device(config-route-map)# set metric 15
```

The following example redistributes OSPF intra-area and interarea routes with next hop devices on serial interface 0/0/0 into the Border Gateway Protocol (BGP) with an INTER_AS metric of 5:

```
Device(config)# router bgp 50000
Device(config-router)# redistribute ospf 1 route-map 10
Device(config-router)# exit
Device(config)# route-map 10 permit
Device(config-route-map)# match route-type internal
Device(config-route-map)# match interface serial 0/0/0
Device(config-route-map)# set metric 5
```

The following example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first type is OSPF external IP routes with tag 5; these routes are inserted into Level 2 IS-IS link-state packets (LSPs) with a metric of 5. The second type is ISO-IGRP derived CLNS prefix routes that match CLNS access list 2000; these routes will be redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
Device(config)# router isis
Device(config-router)# redistribute ospf 1 route-map 2
Device(config-router)# redistribute iso-igrp nsfnet route-map 3

Device(config-router)# exit
Device(config)# route-map 2 permit
Device(config-route-map)# match route-type external
Device(config-route-map)# match tag 5
Device(config-route-map)# set metric 5
Device(config-route-map)# set level level-2
Device(config-route-map)# exit
Device(config)# route-map 3 permit
Device(config-route-map)# match address 2000
Device(config-route-map)# set metric 30
Device(config-route-map)# exit
```

With the following configuration, OSPF external routes with tags 1, 2, 3, and 5 are redistributed into RIP with metrics of 1, 1, 5, and 5, respectively. The OSPF routes with a tag of 4 are not redistributed.

```
Device(config)# router rip
Device(config-router)# redistribute ospf 101 route-map 1
Device(config-router)# exit
Device(config)# route-map 1 permit
Device(config-route-map)# match tag 1 2
Device(config-route-map)# set metric 1
Device(config-route-map)# exit
Device(config)# route-map 1 permit
Device(config-route-map)# match tag 3
Device(config-route-map)# set metric 5
Device(config-route-map)# exit
Device(config)# route-map 1 deny
Device(config-route-map)# match tag 4
Device(config-route-map)# exit
Device(config)# route map 1 permit
Device(config-route-map)# match tag 5
Device(config-route-map)# set metric 5
Device(config-route-map)# exit
```

Given the following configuration, a RIP learned route for network 172.18.0.0 and an ISO-IGRP learned route with prefix 49.0001.0002 will be redistributed into an IS-IS Level 2 LSP with a metric of 5:

```
Device(config)# router isis
```

```
Device(config-router)# redistribute rip route-map 1
Device(config-router)# redistribute iso-igrp remote route-map 1
Device(config-router)# exit
Device(config)# route-map 1 permit
Device(config-route-map)# match ip address 1
Device(config-route-map)# match clns address 2
Device(config-route-map)# set metric 5
Device(config-route-map)# set level level-2
Device(config-route-map)# exit
Device(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Device(config)# clns filter-set 2 permit 49.0001.0002...
```

The following configuration example illustrates how a route map is referenced by the **default-information** router configuration command. This type of reference is called conditional default origination. OSPF will originate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.20.0.0 is in the routing table.

```
Device(config)# route-map ospf-default permit
Device(config-route-map)# match ip address 1
Device(config-route-map)# set metric 5
Device(config-route-map)# set metric-type type-2
Device(config-route-map)# exit
Device(config)# access-list 1 172.20.0.0 0.0.255.255
Device(config)# router ospf 101
Device(config-router)# default-information originate route-map ospf-default
```

Examples: Key Management

The following example configures a key chain named chain1. In this example, the software always accepts and sends key1 as a valid key. The key key2 is accepted from 1:30 p.m. to 3:30 p.m. and is sent from 2:00 p.m. to 3:00 p.m. The overlap allows for migration of keys or discrepancy in the set time of the device. Likewise, the key key3 immediately follows key2, and there is 30-minutes on each side to handle time-of-day differences.

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip rip authentication key-chain chain1
Device(config-if)# ip rip authentication mode md5
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 172.19.0.0
Device(config-router)# version 2
Device(config-router)# exit
Device(config)# key chain chain1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string key1
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string key2
Device(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 2005 duration 7200
Device(config-keychain-key)# send-lifetime 14:00:00 Jan 25 2005 duration 3600
Device(config-keychain-key)# exit
Device(config-keychain)# key 3
Device(config-keychain-key)# key-string key3
Device(config-keychain-key)# accept-lifetime 14:30:00 Jan 25 2005 duration 7200
Device(config-keychain-key)# send-lifetime 15:00:00 Jan 25 2005 duration 3600
Device(config-keychain-key)# end
```

The following example configures a key chain named chain1:

```
Device(config)# key chain chain1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string key1
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string key2
Device(config-keychain-key)# accept-lifetime 00:00:00 Dec 5 2004 23:59:59 Dec 5 2005
Device(config-keychain-key)# send-lifetime 06:00:00 Dec 5 2004 18:00:00 Dec 5 2005
Device(config-keychain-key)# exit
Device(config-keychain)# exit
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 172.19.104.75 255.255.255.0 secondary 172.19.232.147
255.255.255.240
Device(config-if)# ip rip authentication key-chain chain1
Device(config-if)# media-type 10BaseT
Device(config-if)# exit
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# no ip address
Device(config-if)# shutdown
Device(config-if)# media-type 10BaseT
Device(config-if)# exit
Device(config)# interface Fddi 0
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# no keepalive
Device(config-if)# exit
Device(config)# interface Fddi 1/0/0
Device(config-if)# ip address 172.16.1.1 255.255.255.0
Device(config-if)# ip rip send version 1
Device(config-if)# ip rip receive version 1
Device(config-if)# no keepalive
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# version 2
Device(config-router)# network 172.19.0.0
Device(config-router)# network 10.0.0.0
Device(config-router)# network 172.16.0.0
```

Additional References

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference

Technical Assistance

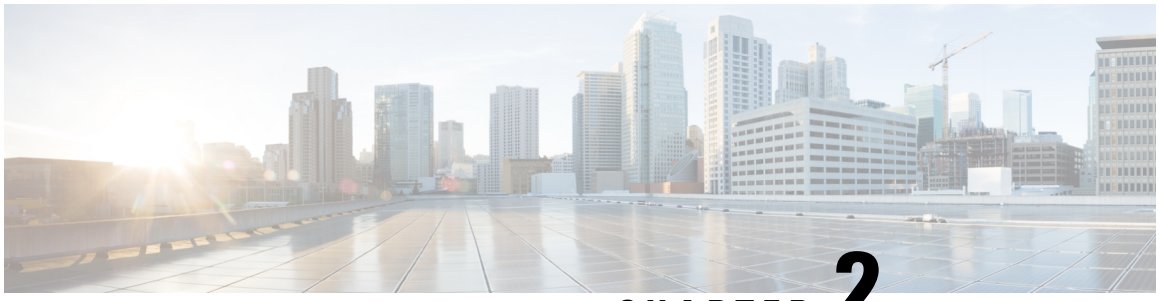
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Basic IP Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Basic IP Routing



CHAPTER 2

IPv6 Routing: Static Routing

This feature provides static routing for IPv6. Static routes are manually configured and define an explicit path between two networking devices.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for IPv6 Routing: Static Routing

Before configuring the device with a static IPv6 route, you must enable the forwarding of IPv6 packets using the **ipv6 unicast-routing** global configuration command, enable IPv6 on at least one interface, and configure an IPv6 address on that interface.

Restrictions for IPv6 Routing: Static Routing

- IPv6 static routes do not support the tag and permanent keywords of the IPv4 **ip route** command.
- IPv6 does not support inserting static routes into virtual routing and forwarding (VRF) tables.

- You should not configure static configurations over dynamic interfaces, because static configurations will be lost during reboot or when the user disconnects and reconnects the device.

Information About IPv6 Routing: Static Routing

Static Routes

Networking devices forward packets using route information that is either manually configured or dynamically learned using a routing protocol. Static routes are manually configured and define an explicit path between two networking devices. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured if the network topology changes. The benefits of using static routes include security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols and no CPU cycles are used to calculate and communicate routes. The main disadvantage to using static routes is the lack of automatic reconfiguration if the network topology changes.

Static routes can be redistributed into dynamic routing protocols but routes generated by dynamic routing protocols cannot be redistributed into the static routing table. No algorithm exists to prevent the configuration of routing loops that use static routes.

Static routes are useful for smaller networks with only one path to an outside network and to provide security for a larger network for certain types of traffic or links to other networks that need more control. In general, most networks use dynamic routing protocols to communicate between networking devices but may have one or two static routes configured for special cases.

Directly Attached Static Routes

In directly attached static routes, only the output interface is specified. The destination is assumed to be directly attached to this interface, so the packet destination is used as the next-hop address. This example shows such a definition:

```
ipv6 route 2001:DB8::/32 gigabitethernet1/0/0
```

The example specifies that all destinations with address prefix 2001:DB8::/32 are directly reachable through interface GigabitEthernet1/0/0.

Directly attached static routes are candidates for insertion in the IPv6 routing table only if they refer to a valid IPv6 interface; that is, an interface that is both up and has IPv6 enabled on it.

Recursive Static Routes

In a recursive static route, only the next hop is specified. The output interface is derived from the next hop. This example shows such a definition:

```
ipv6 route 2001:DB8::/32 2001:DB8:3000:1
```

This example specifies that all destinations with address prefix 2001:DB8::/32 are reachable via the host with address 2001:DB8:3000:1.

A recursive static route is valid (that is, it is a candidate for insertion in the IPv6 routing table) only when the specified next hop resolves, either directly or indirectly, to a valid IPv6 output interface, provided the route does not self-recurse, and the recursion depth does not exceed the maximum IPv6 forwarding recursion depth.

A route self-recurses if it is itself used to resolve its own next hop. For example, suppose we have the following routes in the IPv6 routing table:

```
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
R   2001:DB8::/32 [130/0]
     via ::, Serial2/0
B   2001:DB8:3000:0/16 [200/45]
     Via 2001:DB8::0104
```

The following examples defines a recursive IPv6 static route:

```
ipv6 route
2001:DB8::/32 2001:0BD8:3000:1
```

This static route will not be inserted into the IPv6 routing table because it is self-recursive. The next hop of the static route, 2001:DB8:3000:1, resolves via the BGP route 2001:DB8:3000:0/16, which is itself a recursive route (that is, it only specifies a next hop). The next hop of the BGP route, 2001:DB8::0104, resolves via the static route. Therefore, the static route would be used to resolve its own next hop.

It is not normally useful to manually configure a self-recursive static route, although it is not prohibited. However, a recursive static route that has been inserted in the IPv6 routing table may become self-recursive as a result of some transient change in the network learned through a dynamic routing protocol. If this occurs, the fact that the static route has become self-recursive will be detected and it will be removed from the IPv6 routing table, although not from the configuration. A subsequent network change may cause the static route to no longer be self-recursive, in which case it will be reinserted in the IPv6 routing table.

Fully Specified Static Routes

In a fully specified static route, both the output interface and the next hop are specified. This form of static route is used when the output interface is a multi-access one and it is necessary to explicitly identify the next hop. The next hop must be directly attached to the specified output interface. The following example shows a definition of a fully specified static route:

```
ipv6 route 2001:DB8:/32 gigabitethernet1/0/0 2001:DB8:3000:1
```

A fully specified route is valid (that is, a candidate for insertion into the IPv6 routing table) when the specified IPv6 interface is IPv6-enabled and up.

Floating Static Routes

Floating static routes are static routes that are used to back up dynamic routes learned through configured routing protocols. A floating static route is configured with a higher administrative distance than the dynamic routing protocol it is backing up. As a result, the dynamic route learned through the routing protocol is always used in preference to the floating static route. If the dynamic route learned through the routing protocol is lost, the floating static route will be used in its place. The following example defines a floating static route:

```
ipv6 route 2001:DB8:/32 gigabitethernet1/0/0 2001:DB8:3000:1 210
```

Any of the three types of IPv6 static routes can be used as a floating static route. A floating static route must be configured with an administrative distance that is greater than the administrative distance of the dynamic routing protocol, because routes with smaller administrative distances are preferred.

Note By default, static routes have smaller administrative distances than dynamic routes, so static routes will be used in preference to dynamic routes.

How to Configure IPv6 Static Routing

Configuring a Static IPv6 Route

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix / prefix-length ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast**| **multicast**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** *ipv6-prefix / prefix-length ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast**| **multicast**] [**tag** *tag*] **Example:** `Device(config)# ipv6 route ::/0 serial 2/0`	Configures a static IPv6 route. • A static default IPv6 route is being configured on a serial interface. • See the syntax examples that immediately follow this table for specific uses of the **ipv6 route** command for configuring static routes.

Configuring a Recursive IPv6 Static Route to Use a Default IPv6 Static Route

By default, a recursive IPv6 static route will not resolve using the default route (::/0). Perform this task to restore legacy behavior and allow resolution using the default route.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static resolve default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route static resolve default** **Example:** `Device(config)# ipv6 route static resolve default`	Allows a recursive IPv6 static route to resolve using the default IPv6 static route.

Configuring a Floating Static IPv6 Route

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix* / *prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*} [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** *ipv6-prefix / prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [**tag** *tag*] **Example:** `Device(config)# ipv6 route 2001:DB8::/32 serial 2/0 201`	Configures a static IPv6 route. • In this example, a floating static IPv6 route is being configured. • Default administrative distances are as follows: • Connected interface--0 • Static route--1 • Enhanced Interior Gateway Routing Protocol (EIGRP) summary route--5 • External Border Gateway Protocol (eBGP)--20 • Internal Enhanced IGRP--90 • IGRP--100 • Open Shortest Path First--110 • Intermediate System-to-Intermediate System (IS-IS)--115 • Routing Information Protocol (RIP)--120 • Exterior Gateway Protocol (EGP)--140 • EIGRP external route--170 • Internal BGP--200 • Unknown--255

Verifying Static IPv6 Route Configuration and Operation

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **show ipv6 static** [*ipv6-address* | *ipv6-prefix / prefix-length*][**interface** *interface-type interface-number*] [**recursive**] [**detail**]
   - **show ipv6 route** [*ipv6-address* | *ipv6-prefix / prefix-length* | *protocol* | *interface-type interface-number*]
3. **debug ipv6 routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	

Step 2	Do one of the following:
	• **show ipv6 static** [*ipv6-address* | *ipv6-prefix* / *prefix-length*][**interface** *interface-type interface-number*] [**recursive**] [**detail**]
	• **show ipv6 route** [*ipv6-address* | *ipv6-prefix* / *prefix-length* | *protocol* | *interface-type interface-number*]
	Example:
	`Device# show ipv6 static`
	Example:
	`Device# show ipv6 route static`
		Displays the current contents of the IPv6 routing table.
		• These examples show two different ways of displaying IPv6 static routes.

Step 3	**debug ipv6 routing** **Example:** `Device# debug ipv6 routing`	Displays debugging messages for IPv6 routing table updates and route cache updates.

Configuration Examples for IPv6 Static Routing

Static routes may be used for a variety of purposes. Common usages include the following:

- Manual summarization
- Traffic discard
- Fixed default route
- Backup route

In many cases, alternative mechanisms exist within Cisco software to achieve the same objective. Whether to use static routes or one of the alternative mechanisms depends on local circumstances.

Example Configuring Manual Summarization

The following example shows a static route being used to summarize local interface prefixes advertised into RIP. The static route also serves as a discard route, discarding any packets received by the router to a 2001:DB8:1::/48 destination not covered by a more specific interface prefix.

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet0/0/0
Router(config-if)# ipv6 address 2001:DB8:2:1234/64
```

```
Router(config-if)# exit
Router(config)#
Router(config)# interface gigabitethernet1/0/0
Router(config-if)# ipv6 address 2001:DB8:3:1234/64
Router(config-if)# exit
Router(config)#
Router(config)# interface gigabitethernet2/0/0
Router(config-if)# ipv6 address 2001:DB8:4:1234/64
Router(config-if)# exit
Router(config)#
Router(config)# interface gigabitethernet3/0/0
Router(config-if)# ipv6 address 2001:DB8::1234/64
Router(config-if)# ipv6 rip one enable
Router(config-if)# exit
Router(config)#
Router(config)# ipv6 router rip one
Router(config-rtr)# redistribute static
Router(config-rtr)# exit
Router(config)#
Router(config)# ipv6 route 2001:DB8:1:1/48 null0
Router(config)# end
Router#
00:01:30: %SYS-5-CONFIG_I: Configured from console by console
Router# show ipv6 route static

IPv6 Routing Table - 3 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2001:DB8:1::/48 [1/0]
     via ::, Null0
```

Example: Configuring Traffic Discard

Configuring a static route to point at interface null0 may be used for discarding traffic to a particular prefix. For example, if it is required to discard all traffic to prefix 2001:DB8:42:1/64, the following static route would be defined:

```
Device> enable
Device# configure
 terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ipv6 route 2001:DB8:42:1::/64 null0
Device(config)# end
```

Example: Configuring a Fixed Default Route

A default static route is often used in simple router topologies. In the following example, a router is connected to its local site via GigabitEthernet 0/0/0 and to the main corporate network via Serial 2/0/0 and Serial 3/0/0. All nonlocal traffic will be routed over the two serial interfaces.

```
Router(config)# interface gigabitethernet0/0/0
Router(config-if)# ipv6 address 2001:DB8:17:1234/64
Router(config-if)# exit
Router(config)# interface Serial2/0/0
Router(config-if)# ipv6 address 2001:DB8:1:1234/64
```

```
Router(config-if)# exit
Router(config)# interface Serial3/0/0
Router(config-if)# ipv6 address 2001:DB8:2:124/64
Router(config-if)# exit
Router(config)# ipv6 route ::/0 Serial2/0
Router(config)# ipv6 route ::/0 Serial3/0
Router(config)# end
Router#
00:06:30: %SYS-5-CONFIG_I: Configured from console by console
Router# show ipv6 route static
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
     via ::, Serial2/0
     via ::, Serial3/0
```

Example: Configuring a Floating Static Route

A floating static route often is used to provide a backup path in the event of connectivity failure. In the following example, the router has connectivity to the network core via GigabitEthernet0/0/0 and learns the route 2001:DB8:1:1/32 via IS-IS. If the GigabitEthernet0/0/0 interface fails, or if route 2001:DB8:1:1/32 is no longer learned via IS-IS (indicating loss of connectivity elsewhere in the network), traffic is routed via the backup ISDN interface.

```
Router> enable
Router# configure
 terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet0/0/0
Router(config-if)# ipv6 address 2001:DB8:17:1234/64
Router(config-if)# exit
Router(config)# interface gigabitethernet0/0/0
Router(config-if)# ipv6 address 2001:DB8:1:1234/64
Router(config-if)# ipv6
 router
 isis
Router(config-if)# exit
Router(config)# router isis
Router(config-router)# net 42.0000.0000.0000.0001.00
Router(config-router)# exit
Router(config)# interface BRI1/0
Router(config-if)# encapsulation ppp
Router(config-if)# ipv6 enable
Router(config-if)# isdn switch-type basic-net3
Router(config-if)# ppp authentication chap optional
Router(config-if)# ppp multilink
Router(config-if)# exit
Router(config)# dialer-list 1 protocol ipv6 permit
Router(config)# ipv6 route 2001:DB8:1::/32 BRI1/0 200
Router(config)# end
Router#
00:03:07: %SYS-5-CONFIG_I: Configured from console by console
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

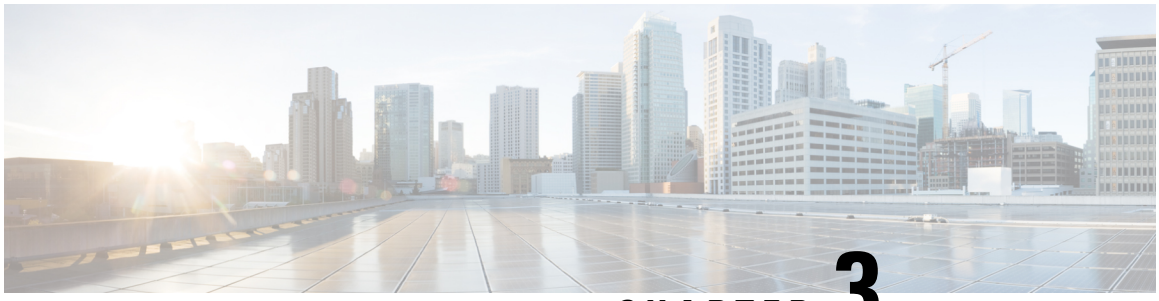
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: Static Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3: Feature Information for IPv6 Routing: Static Routing



CHAPTER 3

Configuring IP Routing Protocol-Independent Features

This module describes how to configure IP routing protocol-independent features. Some of the features discussed in this module include the Default Passive Interface, Fast-Switched Policy Routing, and Policy-Based Routing.

Information About Basic IP Routing

Variable-Length Subnet Masks

Dynamic routing protocols, such as the Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP) Version 2, and static routes support variable-length subnet masks (VLSMs). VLSM enables an organization to use more than one subnet mask within the same network address space. VLSM allows you to conserve IP addresses and efficiently use the available address space. Implementing VLSM is often referred to as "subnetting a subnet."

Note You may want to carefully consider the use of VLSMs. It is easy to make mistakes during address assignments and difficult to monitor networks that use VLSMs. The best way to implement VLSMs is to keep your existing addressing plan in place and gradually migrate some networks to VLSMs to recover address space.

The following example uses two different subnet masks for the class B network address of 172.16.0.0. A subnet mask of /24 is used for LAN interfaces. The /24 mask allows 256 subnets with 254 host IP addresses on each subnet. The final subnet of the range of possible subnets using a /24 subnet mask (172.16.255.0) is reserved for use on point-to-point interfaces and assigned a longer mask of /30. The use of a /30 mask on 172.16.255.0 creates 64 subnets (172.16.255.0–72.16.255.252) with 2 host addresses on each subnet.

Note To ensure unambiguous routing, you must not assign 172.16.255.0/24 to a LAN interface in your network.

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Serial 0/0
Router(config-if)# ip address 172.16.255.5 255.255.255.252
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 172.16.0.0
```

Static Routes

Static routes are user-defined routes that cause packets moving between a source and a destination to take a specified path. Static routes can be important if the router cannot build a route to a particular destination. They are also useful in specifying a gateway of last resort to which all unroutable packets will be sent.

To configure a static route, use the **ip route** command in global configuration mode.

Static routes remain in the router configuration until you remove them (by using the **no** form of the **ip route** command). However, you can override static routes with dynamic routing information through the assignment of administrative distance values. An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers.

Each dynamic routing protocol has a default administrative distance, as listed in the table below. For a configured static route to be overridden, the administrative distance of the static route should be higher than that of the dynamic routing protocol.

Table 4: Default Administrative Distances

Route Source	Default Administrative Distance
Connected interface	0
Static route	1
EIGRP summary route	5
External Border Gateway Protocol (BGP)	20
Internal EIGRP	90
Interior Gateway Routing Protocol (IGRP)	100
OSPF	110
IS-IS	115
RIP	120
Exterior Gateway Protocol (EGP)	140
On-Demand Routing (ODR)	160

Route Source	Default Administrative Distance
External EIGRP	170
Internal BGP	200
Unknown	255

Static routes that point to an interface are advertised through dynamic routing protocols, regardless of whether **redistribute static** router configuration commands were specified for those routing protocols. Static routes that point to an interface are advertised because the routing table considers these routes as connected routes and hence, these routes lose their static nature. However, if you define a static route to an interface that is not connected to one of the networks defined by a **network** command, no dynamic routing protocol will advertise the route unless a **redistribute static** command is specified for the protocols.

When an interface goes down, all static routes associated with that interface are removed from the IP routing table. Also, when the software can no longer find a valid next hop for the address specified as the address of the forwarding router in a static route, the static route is removed from the IP routing table.

Default Routes

Default routes, also known as gateways of last resort, are used to route packets that are addressed to networks not explicitly listed in the routing table. A device might not be able to determine routes to all networks. To provide complete routing capability, network administrators use some devices as smart devices and give the remaining devices default routes to the smart device. (Smart devices have routing table information for the entire internetwork.) Default routes can be either passed along dynamically or configured manually into individual devices.

Most dynamic interior routing protocols include a mechanism for causing a smart device to generate dynamic default information, which is then passed along to other devices.

You can configure a default route by using the following commands:

- **ip default-gateway**
- **ip default-network**
- **ip route 0.0.0.0 0.0.0.0**

You can use the **ip default-gateway** global configuration command to define a default gateway when IP routing is disabled on a device. For instance, if a device is a host, you can use this command to define a default gateway for the device. You can also use this command to transfer a Cisco software image to a device when the device is in boot mode. In boot mode, IP routing is not enabled on the device.

Unlike the **ip default-gateway** command, the **ip default-network** command can be used when IP routing is enabled on a device. When you specify a network by using the **ip default-network** command, the device considers routes to that network for installation as the gateway of last resort on the device.

Gateways of last resort configured by using the **ip default-network** command are propagated differently depending on which routing protocol is propagating the default route. For Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) to propagate the default route, the network specified by the **ip default-network** command must be known to IGRP or EIGRP. The network must be an IGRP- or EIGRP-derived network in the routing table, or the static route used to generate the route to the network must be redistributed into IGRP or EIGRP or advertised into these protocols by using the **network** command. The Routing Information Protocol (RIP) advertises a route to network 0.0.0.0 if a gateway of last

resort is configured by using the **ip default-network** command. The network specified in the **ip default-network** command need not be explicitly advertised under RIP.

Creating a static route to network 0.0.0.0 0.0.0.0 by using the **ip route 0.0.0.0 0.0.0.0** command is another way to set the gateway of last resort on a device. As with the **ip default-network** command, using the static route to 0.0.0.0 is not dependent on any routing protocols. However, IP routing must be enabled on the device. IGRP does not recognize a route to network 0.0.0.0. Therefore, it cannot propagate default routes created by using the **ip route 0.0.0.0 0.0.0.0** command. Use the **ip default-network** command to have IGRP propagate a default route.

EIGRP propagates a route to network 0.0.0.0, but the static route must be redistributed into the routing protocol.

Depending on your release of the Cisco software, the default route created by using the **ip route 0.0.0.0 0.0.0.0** command is automatically advertised by RIP devices. In some releases, RIP does not advertise the default route if the route is not learned via RIP. You might have to redistribute the route into RIP by using the **redistribute** command.

Default routes created using the **ip route 0.0.0.0 0.0.0.0** command are not propagated by Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). Additionally, these default routes cannot be redistributed into OSPF or IS-IS by using the **redistribute** command. Use the **default-information originate** command to generate a default route into an OSPF or IS-IS routing domain.

Default Network

Default networks are used to route packets to destinations not established in the routing table. You can use the **ip default-network** *network-number* global configuration command to configure a default network when IP routing is enabled on the device. When you configure a default network, the device considers routes to that network for installation as the gateway of last resort on the device.

Gateway of Last Resort

When default information is being passed along through a dynamic routing protocol, no further configuration is required. The system periodically scans its routing table to choose the optimal default network as its default route. In the case of RIP, there is only one choice, network 0.0.0.0. In the case of EIGRP, there might be several networks that can be candidates for the system default. Cisco IOS software uses both the administrative distance and metric information to determine the default route (gateway of last resort). The selected default route appears in the gateway of last resort display of the **show ip route** command.

If dynamic default information is not being passed to the software, candidates for the default route are specified with the **ip default-network** global configuration command. In this usage, the **ip default-network** command takes an unconnected network as an argument. If this network appears in the routing table from any source (dynamic or static), the network is flagged as a candidate default route and is a possible choice as the default route.

If the router has no interface on the default network, but does have a route to the default network, the router considers this network as a candidate default path. The route candidates are examined and the best one is chosen based on the administrative distance and metric information. The gateway to the best default path becomes the gateway of last resort.

Maximum Number of Paths

By default, most IP routing protocols install a maximum of four parallel paths in a routing table. Static routes always install six paths. The exception is BGP, which by default allows only one path (the best path) to the destination. However, BGP can be configured to use equal and unequal cost multipath load sharing. See the

"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN" feature in the *BGP Configuration Guide* for more information.

The number of parallel paths that you can configure to be installed in the routing table is dependent on the installed version of the Cisco IOS software. To change the maximum number of parallel paths allowed, use the **maximum-paths** command in router configuration mode.

Multi-Interface Load Splitting

Multi-interface load splitting allows you to efficiently control traffic that travels across multiple interfaces to the same destination. The **traffic-share min** router configuration command specifies that if multiple paths are available to the same destination, only paths with the minimum metric will be installed in the routing table. The number of paths allowed is never more than six. For dynamic routing protocols, the number of paths is controlled by the **maximum-paths** router configuration command. The static route source can install six paths. If more paths are available, the extra paths are discarded. If some installed paths are removed from the routing table, pending routes are added automatically.

Routing Information Redistribution

You can configure the Cisco IOS software to redistribute information from one routing protocol to another. For example, you can configure a device to readvertise EIGRP-derived routes using RIP or to readvertise static routes using EIGRP. Redistribution from one routing protocol to another can be configured in all IP-based routing protocols.

You can also conditionally control the redistribution of routes between routing domains by configuring route maps between two domains. A route map is a route filter that is configured with permit and deny statements, match and set clauses, and sequence numbers. To define a route map for redistribution, use the **route-map** command in global configuration mode.

The metrics of one routing protocol do not necessarily translate into the metrics of another. For example, the RIP metric is hop count and the EIGRP metric is a combination of five metric values. In such situations, a dynamic metric is assigned to the redistributed route. Redistribution in these cases should be applied consistently and carefully in conjunction with inbound filtering to avoid the creation of routing loops.

The following examples illustrate the use of redistribution with and without route maps. The following example shows how to redistribute all OSPF routes into EIGRP:

```
Router(config)# router eigrp 1
Router(config-router)# redistribute ospf 101
Router(config-router)# exit
```

The following example shows how to redistribute RIP routes, with a hop count equal to 1, into OSPF. These routes will be redistributed into OSPF as external LSAs with a metric of 5, metric-type of type 1, and a tag equal to 1.

```
Router(config)# router ospf 1
Router(config-router)# redistribute rip route-map rip-to-ospf
Router(config-router)# exit
Router(config)# route-map rip-to-ospf permit
Router(config-route-map)# match metric 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type 1
Router(config-route-map)# set tag 1
Router(config-route-map)# exit
```

The following example shows how to redistribute OSPF learned routes with tag 7 as a RIP metric of 15:

```
Router(config)# router rip
Router(config-router)# redistribute ospf 1 route-map 5
Router(config-router)# exit
Router(config)# route-map 5 permit
Router(config-route-map)# match tag 7
Router(config-route-map)# set metric 15
```

The following example shows how to redistribute OSPF intra-area and inter-area routes with next-hop routers on serial interface 0/0 into BGP with a metric of 5:

```
Router(config)# router bgp 50000
Router(config-router)# redistribute ospf 1 route-map 10
Router(config-router)# exit
Router(config)# route-map 10 permit
Router(config-route-map)# match route-type internal
Router(config-route-map)# match interface serial 0
Router(config-route-map)# set metric 5
```

The following example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first type is OSPF external IP routes with tag 5; these routes are inserted into Level 2 IS-IS link-state packets (LSPs) with a metric of 5. The second type is ISO-IGRP-derived CLNS prefix routes that match CLNS access list 2000; these routes are redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
Router(config)# router isis
Router(config-router)# redistribute ospf 1 route-map 2
Router(config-router)# redistribute iso-igrp nsfnet route-map 3
Router(config-router)# exit
Router(config)# route-map 2 permit
Router(config-route-map)# match route-type external
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# route-map 3 permit
Router(config-route-map)# match address 2000
Router(config-route-map)# set metric 30
Router(config-route-map)# exit
```

In the following example, OSPF external routes with tags 1, 2, 3, and 5 are redistributed into RIP with metrics of 1, 1, 5, and 5, respectively. The OSPF routes with a tag of 4 are not redistributed.

```
Router(config)# router rip
Router(config-router)# redistribute ospf 101 route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 1 2
Router(config-route-map)# set metric 1
Router(config-route-map)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 3
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
Router(config)# route-map 1 deny
Router(config-route-map)# match tag 4
Router(config-route-map)# exit
Router(config)# route map 1 permit
```

```
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
```

The following example shows how a route map is referenced by using the **default-information** router configuration command. Such referencing is called conditional default origination. OSPF will generate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.16.0.0 is in the routing table.

```
Router(config)# route-map ospf-default permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type-2
Router(config-route-map)# exit
Router(config)# access-list 1 172.16.0.0 0.0.255.255
Router(config)# router ospf 101
Router(config-router)# default-information originate route-map ospf-default
```

Supported Automatic Metric Translations

This section describes supported automatic metric translations between routing protocols. The following points are based on the assumption that you have not defined a default redistribution metric that replaces metric conversions:

- RIP can automatically redistribute static routes. It assigns static routes a metric of 1 (directly connected).
- BGP does not send metrics in its routing updates.
- EIGRP can automatically redistribute static routes from other EIGRP-routed autonomous systems as long as the static route and any associated interfaces are covered by an EIGRP network statement. EIGRP assigns static routes a metric that identifies them as directly connected. EIGRP does not change the metrics of routes derived from EIGRP updates from other autonomous systems.

Note Any protocol can redistribute routes from other routing protocols as long as a default metric is configured.

Protocol Differences in Implementing the no redistribute Command

Caution Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting. In most cases, changing or disabling any keyword will not affect the state of other keywords.

Different protocols implement the **no redistribute** command differently as follows:

- In Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP) configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.

- The **no redistribute isis** command removes the Intermediate System to Intermediate System (IS-IS) redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- The Enhanced Interior Gateway Routing Protocol (EIGRP) used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.

Default Passive Interfaces

The Default Passive Interfaces feature simplifies the configuration of distribution devices by allowing all interfaces to be set as passive by default. In ISPs and large enterprise networks, many distribution devices have more than 200 interfaces. Obtaining routing information from these interfaces requires configuration of the routing protocol on all interfaces and manual configuration of the **passive-interface** command on interfaces where adjacencies were not desired.

Sources of Routing Information Filtering

Filtering sources of routing information prioritizes routing information gathered from different sources because some pieces of routing information may be more accurate than others. An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is an integer from 0 to 255. In general, the higher the value the lower the trust rating. An administrative distance of 255 means that the routing information source cannot be trusted at all and should be ignored.

In a large network, some routing protocols and some routers can be more reliable than others as sources of routing information. Also, when multiple routing processes are running on the same router for IP, the same route may be advertised by more than one routing process. By specifying administrative distance values, you enable a router to intelligently discriminate between sources of routing information. The router will always pick the route whose routing protocol has the lowest administrative distance.

There are no guidelines for assigning administrative distances because each network has its own requirements. You must determine a reasonable matrix of administrative distances for a network as a whole.

Note You can use the administrative distance to rate the routing information from routers that are running the same routing protocol. However, using the administrative distance for this purpose can result in inconsistent routing information and forwarding loops.

In the following example, the **router eigrp** global configuration command configures EIGRP routing in autonomous system 1. The **network** command specifies EIGRP routing on networks 192.0.2.16 and 172.16.0.0. The first **distance** router configuration command sets the default administrative distance to 255, which instructs the router to ignore all routing updates from routers for which an explicit distance has not been set. The second **distance** command sets the administrative distance to 80 for internal EIGRP routes and to 100 for external EIGRP routes. The third **distance** command sets the administrative distance to 120 for the router with the address 172.16.1.3.

```
Router(config)# router eigrp 1
Router(config-router)# network 192.0.2.16
Router(config-router)# network 172.16.0.0
```

```
Router(config-router)# distance 255
Router(config-router)# distance eigrp 80 100
Router(config-router)# distance 120 172.16.1.3 0.0.0.0
```

Note The **distance eigrp** command must be used to set the administrative distance for EIGRP-derived routes.

The following example assigns the router with the address 192.0.2.1 an administrative distance of 100 and all other routers on subnet 192.0.2.0 an administrative distance of 200:

```
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
```

However, if you reverse the order of these two commands, all routers on subnet 192.0.2.0 are assigned an administrative distance of 200, including the router at address 192.0.2.1:

```
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
```

Note Administrative distances should be applied carefully and consistently to avoid the creation of routing loops or other network failures.

In the following example, the administrative distance value for learned IP routes is 90. Preference is given to these IP routes rather than routes with the default administrative distance value of 110.

```
Router(config)# router isis
Router(config-router)# distance 90 ip
```

Policy-Based Routing

Policy-based routing (PBR) is a more flexible mechanism than destination routing for routing packets. It is a process whereby a router puts packets through a route map before routing them. The route map determines which packets are routed to which router next. You can enable PBR if you want certain packets to be routed some way other than the obvious shortest path. Possible applications for policy-based routing include protocol-sensitive routing, source-sensitive routing, routing based on interactive versus batch traffic, and routing based on dedicated links.

To enable PBR, you must identify the route map to be used for PBR and create the route map. The route map specifies the match criteria and the resulting action if all match clauses are met.

A packet arriving on a specified interface will be subject to PBR, except when its destination IP address is the same as the IP address of the router's interface. To disable fast switching of all packets arriving on this interface, use the **ip policy route-map** command in interface configuration mode.

To define the route map to be used for PBR, use the **route-map** command in global configuration mode.

To define the criteria by which packets are examined to learn if they will follow PBR, use either the **match length** command or the **match ip address** command or both in route map configuration mode. The **match length** command allows you to configure policy routing based on the Level 3 length of the packet, and the

match ip address command allows you to policy route packets based on the criteria that can be matched with an extended access list.

The following example provides two sources with equal access to two different service providers. Packets that arrive on asynchronous interface 1 from the source 10.1.1.1 are sent to the router at 172.16.6.6 if the router has no explicit route for the destination of the packets. Packets that arrive from the source 172.17.2.2 are sent to the router at 192.168.7.7 if the router has no explicit route for the destination of the packets. All other packets for which the router has no explicit route to the destination are discarded.

```
Router(config)# access-list 1 permit ip 10.1.1.1
Router(config)# access-list 2 permit ip 172.17.2.2
Router(config)# interface async 1
Router(config-if)# ip policy route-map equal-access
Router(config-if)# exit
Router(config)# route-map equal-access permit 10
Router(config-route-map)# match ip address 1
Router(config-route-map)# set ip default next-hop 172.16.6.6
Router(config-route-map)# exit
Router(config)# route-map equal-access permit 20
Router(config-route-map)# match ip address 2
Router(config-route-map)# set ip default next-hop 192.168.7.7
Router(config-route-map)# exit
Router(config)# route-map equal-access permit 30
Router(config-route-map)# set default interface null 0
Router(config-route-map)# exit
```

You can set IP header precedence bits in the router when PBR is enabled. The precedence setting in the IP header determines how packets are treated during times of high traffic. When packets containing these headers arrive at another router, the packets are ordered for transmission according to the precedence set if the queuing feature is enabled. The router does not honor the precedence bits if queuing is not enabled, and the packets are sent in FIFO order. You can change the precedence setting by using either a number or a name.

The table below lists the possible IP Precedence values (numbers and their corresponding names), from the least important to the most important.

Table 5: IP Precedence Values

Number	Name
0	routine
1	priority
2	immediate
3	flash
4	flash-override
5	critical
6	internet
7	network

Fast-Switched Policy Routing

IP policy routing can be fast-switched. Prior to fast-switched policy routing, policy routing could only be process -switched, which meant that on most platforms, the switching rate was approximately 1000 to 10,000 packets per second. Such rates were not fast enough for many applications. With fast-switched policy routing, users who need policy routing to occur at faster speeds can implement policy routing without slowing down the device.

Fast-switched policy routing supports all **match** commands and most **set** commands, except for the following:

- **set ip default**
- **set interface**

The **set interface** command is supported only over point-to-point links, unless there is a route cache entry that uses the same interface that is specified in the command in the route map.

To configure fast-switched policy routing, use the **ip route-cache policy** interface configuration command.

Local Policy Routing

Packets that are generated by the router are not normally policy-routed. To enable local policy routing for such packets, you must indicate which route map the router should use. All packets originating on the router will then be subject to local policy routing. To identify the route map to be used for local policy routing, use the **ip local policy route-map** command in global configuration mode.

Use the **show ip local policy** command to display the route map used for local policy routing, if one exists.

NetFlow Policy Routing

NetFlow policy routing (NPR) integrates policy routing, which enables traffic engineering and traffic classification, with NetFlow services, which provide billing, capacity planning, and information monitoring on real-time traffic flows. IP policy routing works with Cisco Express Forwarding (formerly known as CEF), distributed Cisco Express Forwarding (formerly known as dCEF), and NetFlow.

NetFlow policy routing leverages the following technologies:

- Cisco Express Forwarding, which looks at a Forwarding Information Base (FIB) instead of a routing table when switching packets, to address maintenance problems of a demand caching scheme.
- Distributed Cisco Express Forwarding, which addresses the scalability and maintenance problems of a demand caching scheme.
- NetFlow, which provides accounting, capacity planning, and traffic monitoring capabilities.

The following are the benefits of NPR:

- NPR takes advantage of new switching services. Cisco Express Forwarding, distributed Cisco Express Forwarding, and NetFlow can now use policy routing.
- Policy routing can be deployed on a wide scale and on high-speed interfaces.

NPR is the default policy routing mode. No additional configuration tasks are required to enable policy routing with Cisco Express Forwarding, distributed Cisco Express Forwarding, or NetFlow. As soon as one of these features is turned on, packets are automatically subjected to policy routing in the appropriate switching path.

The following example shows how to configure policy routing with Cisco Express Forwarding. The route is configured to verify that the next hop 10.0.0.8 of the route map named test is a Cisco Discovery Protocol neighbor before the device tries to policy-route to it.

```
Device(config)# ip cef
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip route-cache flow
Device(config-if)# ip policy route-map test
Device(config-if)# exit
Device(config)# route-map test permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip precedence priority
Device(config-route-map)# set ip next-hop 10.0.0.8
Device(config-route-map)# set ip next-hop verify-availability
Device(config-route-map)# exit
Device(config)# route-map test permit 20
Device(config-route-map)# match ip address 101
Device(config-route-map)# set interface Ethernet 0/0/3
Device(config-route-map)# set ip tos max-throughput
Device(config-route-map)# exit
```

Authentication Key Management and Supported Protocols

Key management is a method of controlling the authentication keys used by routing protocols. Not all protocols support key management. Authentication keys are available for Director Response Protocol (DRP) Agent, Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP) Version 2.

You can manage authentication keys by defining key chains, identifying the keys that belong to the key chain, and specifying how long each key is valid. Each key has its own key identifier (specified using the **key chain** configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and the message digest algorithm 5 (MD5) authentication key in use.

You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes.

How to Configure Basic IP Routing

Redistributing Routing Information

You can redistribute routes from one routing domain into another, with or without controlling the redistribution with a route map. To control which routes are redistributed, configure a route map and reference the route map from the **redistribute** command.

The tasks in this section describe how to define the conditions for redistributing routes (a route map), how to redistribute routes, and how to remove options for redistributing routes, depending on the protocol being used.

Defining Conditions for Redistributing Routes

Route maps can be used to control route redistribution (or to implement policy-based routing). To define conditions for redistributing routes from one routing protocol into another, configure the **route-map** command. Then use at least one **match** command in route map configuration mode, as needed. At least one **match** command is used in this task because the purpose of the task is to illustrate how to define one or more conditions on which to base redistribution.

Note A route map is not required to have **match** commands; it can have only **set** commands. If there are no **match** commands, everything matches the route map.

Note There are many more **match** commands not shown in this table. For additional **match** commands, see the *Cisco IOS Master Command List*.

Command or Action	Purpose
match as-path *path-list-number*	Matches a BGP autonomous system path access list.
match community {*standard-list-number* | *expanded-list-number* | *community-list-name* **match community**[**exact**]}	Matches a BGP community.
match ip address {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}	Matches routes that have a destination network address that is permitted to policy route packets or is permitted by a standard access list, an extended access list, or a prefix list.
match metric *metric-value*	Matches routes with the specified metric.
match ip next-hop {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches a next-hop device address passed by one of the specified access lists.
match tag *tag-value* [*tag-value*]	Matches the specified tag value.
match interface *type number* [*type number*]	Matches routes that use the specified interface as the next hop.
match ip route-source {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*]	Matches the address specified by the advertised access lists.

Command or Action	Purpose
match route-type {**local** | **internal** | **external** [**type-1** | **type-2**] | **level-1** | **level-2**}	Matches the specified route type.

To optionally specify the routing actions for the system to perform if the match criteria are met (for routes that are being redistributed by the route map), use one or more **set** commands in route map configuration mode, as needed.

Note A route map is not required to have **set** commands; it can have only **match** commands.

Note There are more **set** commands not shown in this table. For additional **set** commands, see the *Cisco IOS Master Command List*.

Command or Action	Purpose
set community {*community-number* [**additive**] [**well-known**]| **none**}	Sets the community attribute (for BGP).
set dampening *halflife reuse suppress max-suppress-time*	Sets route dampening parameters (for BGP).
set local-preference *number-value*	Assigns a local preference value to a path (for BGP).
set origin {**igp** | **egp** *as-number* | **incomplete**}	Sets the route origin code.
set as-path{**tag** | **prepend** *as-path-string* }	Modifies the autonomous system path (for BGP).
set next-hop *next-hop*	Specifies the address of the next hop.
set automatic-tag	Enables automatic computation of the tag table.
set level {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**}	Specifies the areas to import routes.
set metric *metric-value*	Sets the metric value for redistributed routes (for any protocol, except EIGRP).

Command or Action	Purpose
set metric *bandwidth delay reliability load mtu*	Sets the metric value for redistributed routes (for EIGRP only).
set metric-type {**internal** | **external** | **type-1** | **type-2**}	Sets the metric type for redistributed routes.
set metric-type internal	Sets the Multi Exit Discriminator (MED) value on prefixes advertised to the external BGP neighbor to match the Interior Gateway Protocol (IGP) metric of the next hop.
set tag *tag-value*	Sets a tag value to be applied to redistributed routes.

Redistributing Routes from One Routing Domain to Another

Perform this task to redistribute routes from one routing domain into another and to control route redistribution. This task shows how to redistribute OSPF routes into a BGP domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system*
4. **redistribute** *protocol process-id*
5. **default-metric** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 109`	Enables a BGP routing process and enters router configuration mode.

	Command or Action	Purpose
Step 4	**redistribute** *protocol process-id* **Example:** `Device(config-router)# redistribute ospf 2 1`	Redistributes routes from the specified routing domain into another routing domain.
Step 5	**default-metric** *number* **Example:** `Device(config-router)# default-metric 10`	Sets the default metric value for redistributed routes. **Note** The metric value specified in the **redistribute** command supersedes the metric value specified using the **default-metric** command.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Removing Options for Redistribution Routes

Caution Removing options that you have configured for the **redistribute** command requires careful use of the **no redistribute** command to ensure that you obtain the result that you are expecting.

Different protocols implement the **no redistribute** command differently as follows:

- In BGP, OSPF, and RIP configurations, the **no redistribute** command removes only the specified keywords from the **redistribute** commands in the running configuration. They use the *subtractive keyword* method when redistributing from other protocols. For example, in the case of BGP, if you configure **no redistribute static route-map interior**, only the route map is removed from the redistribution, leaving **redistribute static** in place with no filter.
- The **no redistribute isis** command removes the IS-IS redistribution from the running configuration. IS-IS removes the entire command, regardless of whether IS-IS is the redistributed or redistributing protocol.
- EIGRP used the subtractive keyword method prior to EIGRP component version rel5. Starting with EIGRP component version rel5, the **no redistribute** command removes the entire **redistribute** command when redistributing from any other protocol.
- For the **no redistribute connected** command, the behavior is subtractive if the **redistribute** command is configured under the **router bgp** or the **router ospf** command. The behavior is complete removal of the command if it is configured under the **router isis** or the **router eigrp** command.

The following OSPF commands illustrate how various options are removed from the redistribution in router configuration mode.

Command or Action	Purpose
`no redistribute connected metric 1000 subnets`	Removes the configured metric value of 1000 and the configured subnets and retains the **redistribute connected** command in the configuration.
`no redistribute connected metric 1000`	Removes the configured metric value of 1000 and retains the **redistribute connected subnets** command in the configuration.
`no redistribute connected subnets`	Removes the configured subnets and retains the **redistribute connected metric** *metric-value* command in the configuration.
`no redistribute connected`	Removes the **redistribute connected** command and any of the options that were configured for the command.

Configuring Routing Information Filtering

To filter routing protocol information, perform the tasks in this section.

Note When routes are redistributed between OSPF processes, no OSPF metric is preserved.

Preventing Routing Updates Through an Interface

To prevent other routers on a local network from dynamically learning routes, you can keep routing update messages from being sent through a router interface. To prevent routing updates through a specified interface, use the **passive-interface** command in router configuration mode. This command is supported in all IP-based routing protocols, except BGP.

OSPF and IS-IS behave differently. In OSPF, the interface address that you specify as passive appears as a stub network in the OSPF domain. OSPF routing information is neither sent nor received through the specified router interface. In IS-IS, the specified IP addresses are advertised without actually running IS-IS on those interfaces.

Configuring Default Passive Interfaces

Perform this task to set all interfaces on a device, in an Enhanced Interior Gateway Routing Protocol (EIGRP) environment, as passive by default, and then activate only those interfaces where adjacencies are desired.

Note When **passive-interface default** and **no-passive interface <int_name>** are configured, the **show run** command displays both interfaces. If you configure **passive-interface default** again, the **show run** command displays only the **passive-interface default**, and this causes the OSPF neighbors (if any) to flap. This behavior is specific to OSPF, and differs from other IGPs such as EIGRP and IS-IS.

In Cisco IOS XE 17.6.7, 17.9.5, 17.12.3, 17.14.x, and higher releases, this behavior has been modified for OSPF to be in line with EIGRP and IS-IS, i.e., when **passive-interface default** and **no-passive interface <int_name>** are configured, and you configure **passive-interface default** again, the **show run** command displays both interfaces, and OSPF neighbors do not flap.

This update is available on the following platforms:

- Cisco Catalyst 8500L Edge Platforms

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** {*autonomous-system-number* | *virtual-instance-number*}
4. **passive-interface** [**default**] [*type number*]
5. **no passive-interface** [**default**] [*type number*]
6. **network** *network-address* [*options*]
7. **end**
8. **show ip eigrp interfaces**
9. **show ip interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** {*autonomous-system-number* | *virtual-instance-number*} **Example:** `Device(config)# router eigrp 1`	Configures an EIGRP process and enters router configuration mode. • *autonomous-system-number*—Autonomous system number that identifies the services to the other EIGRP address-family devices. It is also used to tag routing information. The range is 1 to 65535.

	Command or Action	Purpose
		• *virtual-instance-number*—EIGRP virtual instance name. This name must be unique among all address-family router processes on a single device, but need not be unique among devices
Step 4	**passive-interface** [**default**] [*type number*] **Example:** `Device(config-router)# passive-interface default`	Sets all interfaces as passive by default.
Step 5	**no passive-interface** [**default**] [*type number*] **Example:** `Device(config-router)# no passive-interface gigabitethernet 0/0/0`	Activates only those interfaces that need adjacencies.
Step 6	**network** *network-address* [*options*] **Example:** `Device(config-router)# network 192.0.2.0`	Specifies the list of networks to be advertised by routing protocols.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 8	**show ip eigrp interfaces** **Example:** `Device# show ip eigrp interfaces`	Verifies whether interfaces on your network have been set to passive.
Step 9	**show ip interface** **Example:** `Device# show ip interface`	Verifies whether interfaces you enabled are active.

Controlling the Advertising of Routes in Routing Updates

To prevent other devices from learning one or more routes, you can suppress routes from being advertised in routing updates. To suppress routes from being advertised in routing updates, use the **distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing-process* | *as-number*] command in router configuration mode.

You cannot specify an interface name in Open Shortest Path First (OSPF). When used for OSPF, this feature applies only to external routes.

Controlling the Processing of Routing Updates

You might want to avoid processing certain routes listed in incoming updates. This feature does not apply to OSPF or IS-IS. To suppress routes in incoming updates, use the **distribute-list** {*access-list-number* | *access-list-name*} **in** [*interface-type interface-number*] command in router configuration mode.

Filtering Sources of Routing Information

To filter sources of routing information, use the **distance** *ip-address wildcard- mask* [*ip-standard-acl* | *ip-extended-acl* | *access-list-name*] command in router configuration mode.

Configuring Precedence for Policy-Based Routing Default Next-Hop Routes

Perform this task to configure the precedence of packets and specify where packets that pass the match criteria are output.

Note The **set ip next-hop** and **set ip default next-hop** commands are similar but have a different order of operation. Configuring the **set ip next-hop** command causes the system to first use policy routing and then use the routing table. Configuring the **set ip default next-hop** command causes the system to first use the routing table and then the policy-route-specified next hop.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
4. **set ip precedence** {*number* | *name*}
5. **set ip next-hop** *ip-address* [*ip-address*]
6. **set interface** *type number* [*...type number*]
7. **set ip default next-hop** *ip-address* [*ip-address*]
8. **set default interface** *type number* [*...type number*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed.
Step 4	**set ip precedence** {*number* | *name*} **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the precedence value in the IP header. **Note** You can specify either a precedence number or a precedence name.
Step 5	**set ip next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ip next-hop 192.0.2.1`	Specifies the next hop for routing packets. **Note** The next hop must be an adjacent device.
Step 6	**set interface** *type number* [*...type number*] **Example:** `Device(config-route-map)# set interface gigabitethernet 0/0/0`	Specifies the output interface for the packet.
Step 7	**set ip default next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ip default next-hop 172.16.6.6`	Specifies the next hop for routing packets if there is no explicit route for this destination. **Note** Like the **set ip next-hop** command, the **set ip default next-hop** command must specify an adjacent device.
Step 8	**set default interface** *type number* [*...type number*] **Example:** `Device(config-route-map)# set default interface serial 0/0/0`	Specifies the output interface for the packet if there is no explicit route for the destination.
Step 9	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation via BGP

Configuring QoS Policy Propagation via BGP Based on Community Lists

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
4. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
5. **set ip precedence** [*number* | *name*]
6. **exit**
7. **router bgp** *autonomous-system*
8. **table-map** *route-map-name*
9. **exit**
10. **ip community-list** *standard-list-number* {**permit** | **deny**} [*community-number*]
11. **interface** *type number*
12. **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. **exit**
14. **ip bgp-community new-format**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. .
Step 4	**match community** {*standard-list-number*	*expanded-list-number*	*community-list-name* [**exact**]} **Example:** `Device(config-route-map)# match community 1`	Matches a Border Gateway Protocol (BGP) community list.

	Command or Action	Purpose
Step 5	**set ip precedence** [*number* | *name*] **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the community list matches. **Note** You can specify either a precedence number or a precedence name.
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 45000`	Enables a BGP process and enters router configuration mode.
Step 8	**table-map** *route-map-name* **Example:** `Device(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 10	**ip community-list** *standard-list-number* {**permit** | **deny**} [*community-number*] **Example:** `Device(config)# ip community-list 1 permit 2`	Creates a community list for BGP and controls access to it.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 12	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Device(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 14	**ip bgp-community new-format** **Example:** `Device(config)# ip bgp-community new-format`	(Optional) Displays the BGP community number in AA:NN (autonomous system:community number/4-byte number) format.
Step 15	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation via BGP Based on the Autonomous System Path Attribute

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **named-ordering-route-map enable**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq** *sequence-name*
5. **match as-path** *path-list-number*
6. **set ip precedence** [*number* | *name*]
7. **exit**
8. **router bgp** *autonomous-system*
9. **table-map** *route-map-name*
10. **exit**
11. **ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression*
12. **interface** *type number*
13. **bgp-policy** {**source** | **destination**} **ip-prec-map**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**named-ordering-route-map enable**] **Example:**	Enables ordering of route-maps based on a string provided by the user.

	Command or Action	Purpose
	`Device(config)# named-ordering-route-map enable`	
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq** *sequence-name* **Example:** `Device(config)# route-map alpha permit ordering-seq sequence1`	Configures a route map and specifies how the packets are to be distributed. **ordering-seq** indicates the sequence that is to be used for ordering of route-maps.
Step 5	**match as-path** *path-list-number* **Example:** `Device(config-route-map)# match as-path 2`	Matches a Border Gateway Protocol (BGP) autonomous system path access list.
Step 6	**set ip precedence** [*number* | *name*] **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the autonomous-system path matches. **Note** You can specify either a precedence number or a precedence name.
Step 7	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 8	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 45000`	Enables a BGP process and enters router configuration mode.
Step 9	**table-map** *route-map-name* **Example:** `Device(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 11	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression* **Example:** `Device(config)# ip as-path access-list 500 permit 45000`	Defines an autonomous system path access list.
Step 12	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.

	Command or Action	Purpose
Step 13	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Device(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation Based on an Access List

This section describes how to configure the QoS Policy Propagation via BGP feature based on an access list. This section assumes that you have already configured Cisco Express Forwarding or distributed Cisco Express Forwarding and BGP on your router.

Perform this task to configure the router to propagate the IP precedence based on an access list:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *route-map-name* [**permit** | **deny** [*sequence-number*]]
4. **match ip address** *access-list-number*
5. **set ip precedence** [*number* | *name*]
6. **exit**
7. **router bgp** *autonomous-system*
8. **table-map** *route-map-name*
9. **exit**
10. **access-list** *access-list-number* {**permit** | **deny**} *source*
11. **interface** *type number*
12. **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**route-map** *route-map-name* [**permit** | **deny** [*sequence-number*]] **Example:** `Router(config)# route-map rm1`	Defines a route map to control redistribution and enters route-map configuration mode.
Step 4	**match ip address** *access-list-number* **Example:** `Router(config-route-map)# match ip address 3`	Matches routes that have a destination network address that is permitted by a standard or extended access list.
Step 5	**set ip precedence** [*number* | *name*] **Example:** `Router(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the autonomous system path matches. **Note** You can specify either a precedence number or a precedence name.
Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system* **Example:** `Router(config)# router bgp 45000`	Enables a BGP routing process and enters router configuration mode.
Step 8	**table-map** *route-map-name* **Example:** `Router(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 10	**access-list** *access-list-number* {**permit** | **deny**} *source* **Example:** `Router(config)# access-list 2 permit 172.16.0.2`	Defines an access list.
Step 11	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 12	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Router(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 13	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-if)# end`	

Monitoring QoS Policy Propagation via BGP

To monitor the QoS Policy Propagation via the BGP feature configuration, use the following optional commands.

Command or Action	Purpose
show ip bgp	Displays entries in the Border Gateway Protocol (BGP) routing table to verify whether the correct community is set on the prefixes.
show ip bgp community-list *community-list-number*	Displays routes permitted by the BGP community to verify whether correct prefixes are selected.
show ip cef *network*	Displays entries in the forwarding information base (FIB) table based on the specified IP address to verify whether Cisco Express Forwarding has the correct precedence value for the prefix.
show ip interface	Displays information about the interface.
show ip route *prefix*	Displays the current status of the routing table to verify whether correct precedence values are set on the prefixes.

Managing Authentication Keys

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *number*
5. **key-string** *text*
6. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
7. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
8. **end**

9. **show key chain**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in ascending order and uses the first valid key it encounters. The lifetimes allow for overlap during key changes. `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain chain1`	Defines a key chain and enters key-chain configuration mode.
Step 4	**key** *number* **Example:** `Device(config-keychain)# key 1`	Identifies number of an authentication key on a key chain. The range of keys is from 0 to 2147483647. The key identification numbers need not be consecutive.
Step 5	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string string1`	Identifies the key string.
Step 6	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# accept-lifetime 13:30:00 Dec 22 2011 duration 7200`	Specifies the time period during which the key can be received.
Step 7	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime 14:30:00 Dec 22 2011 duration 3600`	Specifies the time period during which the key can be sent.
Step 8	**end** **Example:** `Device(config-keychain-key)# end`	Exits key-chain key configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show key chain** **Example:** `Device# show key chain`	(Optional) Displays authentication key information.

Monitoring and Maintaining the IP Network

Clearing Routes from the IP Routing Table

You can remove all contents of a particular table. Clearing a table may become necessary when the contents of the particular structure have become, or are suspected to be, invalid.

To clear one or more routes from the IP routing table, use the **clear ip route** {*network* [*mask*] | *****} command in privileged EXEC mode.

Displaying System and Network Statistics

You can use the following **show** commands to display system and network statistics. You can display specific statistics such as contents of IP routing tables, caches, and databases. You can also display information about node reachability and discover the routing path that packets leaving your device are taking through the network. This information can an be used to determine resource utilization and solve network problems.

Command or Action	Purpose
show ip cache policy	Displays cache entries in the policy route cache.
show ip local policy	Displays the local policy route map if one exists.
show ip policy	Displays policy route maps.
show ip protocols	Displays the parameters and current state of the active routing protocols.
show ip route [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **list** {**access-list-number** | *access-list-name*} | **static download**]	Displays the current state of the routing table.
show ip route summary	Displays the current state of the routing table in summary form.
show ip route supernets-only	Displays supernets.
show key chain [*name-of-chain*]	Displays authentication key information.

Command or Action	Purpose
show route-map [*map-name*]	Displays all route maps configured or only the one specified.

Configuration Examples for Basic IP Routing

Example: Variable-Length Subnet Mask

The following example uses two different subnet masks for the class B network address of 172.16.0.0. A subnet mask of /24 is used for LAN interfaces. The /24 mask allows 256 subnets with 254 host IP addresses on each subnet. The final subnet of the range of possible subnets using a /24 mask (172.16.255.0) is reserved for use on point-to-point interfaces and assigned a longer mask of /30. The use of a /30 mask on 172.16.255.0 creates 64 subnets (172.16.255.0 - 172.16.255.252) with 2 host addresses on each subnet.

Danger To ensure unambiguous routing, you must not assign 172.16.255.0/24 to a LAN interface in your network.

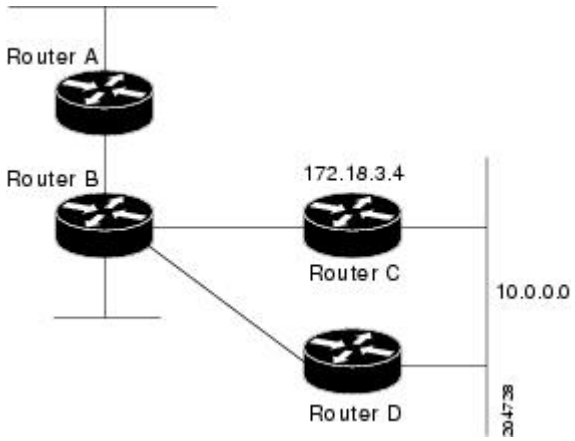
```
Router(config)# interface Ethernet 0/0

Router(config-if)# ip address 172.16.1.1 255.255.255.0

Router(config-if)# ! 8 bits of host address space reserved for Ethernet interfaces
Router(config-if)# exit
Router(config)# interface Serial 0/0

Router(config-if)# ip address 172.16.255.5 255.255.255.252

Router(config-if)# ! 2 bits of address space reserved for point-to-point serial interfaces

Router(config-if)# exit

Router(config)# router rip
Router(config-router)# network 172.16.0.0
Router(config-router)# ! Specifies the network directly connected to the router
```

Example: Overriding Static Routes with Dynamic Protocols

In the following example, packets for network 10.0.0.0 from Router B (where the static route is installed) will be routed through 172.18.3.4 if a route with an administrative distance less than 110 is not available. The figure below illustrates this example. The route learned by a protocol with an administrative distance of less than 110 might cause Router B to send traffic destined for network 10.0.0.0 via the alternate path--through Router D.

```
Router(config)# ip route 10.0.0.0 255.0.0.0 172.18.3.4 110
```

Figure 4: Overriding Static Routes



Example: Administrative Distances

In the following example, the **router eigrp** global configuration command configures EIGRP routing in autonomous system 1. The **network** command specifies EIGRP routing on networks 192.0.2.16 and 172.16.0.0. The first **distance** router configuration command sets the default administrative distance to 255, which instructs the router to ignore all routing updates from routers for which an explicit distance has not been set. The second **distance** command sets the administrative distance to 80 for internal EIGRP routes and to 100 for external EIGRP routes. The third **distance** command sets the administrative distance to 120 for the router with the address 172.16.1.3.

```
Router(config)# router eigrp 1
Router(config-router)# network 192.0.2.16
Router(config-router)# network 172.16.0.0
Router(config-router)# distance 255
Router(config-router)# distance eigrp 80 100
Router(config-router)# distance 120 172.16.1.3 0.0.0.0
```

Note The **distance eigrp** command must be used to set the administrative distance for EIGRP-derived routes.

The following example assigns the router with the address 192.0.2.1 an administrative distance of 100 and all other routers on subnet 192.0.2.0 an administrative distance of 200:

```
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
```

However, if you reverse the order of these two commands, all routers on subnet 192.0.2.0 are assigned an administrative distance of 200, including the router at address 192.0.2.1:

```
Router(config-router)# distance 200 192.0.2.0 0.0.0.255
Router(config-router)# distance 100 192.0.2.1 0.0.0.0
```

Note Assigning administrative distances can be used to solve unique problems. However, administrative distances should be applied carefully and consistently to avoid the creation of routing loops or other network failures.

In the following example, the distance value for learned IP routes is 90. Preference is given to these IP routes rather than routes with the default administrative distance value of 110.

```
Router(config)# router isis

Router(config-router)# distance 90 ip
```

Example: Static Routing Redistribution

In the example that follows, three static routes are specified, two of which are to be advertised. The static routes are created by specifying the **redistribute static** router configuration command and then specifying an access list that allows only those two networks to be passed to the EIGRP process. Any redistributed static routes should be sourced by a single router to minimize the likelihood of creating a routing loop.

```
Router(config)# ip route 192.168.2.0 255.255.255.0 192.168.7.65
Router(config)# ip route 192.168.5.0 255.255.255.0 192.168.7.65
Router(config)# ip route 10.10.10.0 255.255.255.0 10.20.1.2
Router(config)# !
Router(config)# access-list 3 permit 192.168.2.0 0.0.255.255
Router(config)# access-list 3 permit 192.168.5.0 0.0.255.255
Router(config)# access-list 3 permit 10.10.10.0 0.0.0.255
Router(config)# !
Router(config)# router eigrp 1
Router(config-router)# network 192.168.0.0
Router(config-router)# network 10.10.10.0
Router(config-router)# redistribute static metric 10000 100 255 1 1500
Router(config-router)# distribute-list 3 out static
```

Example: EIGRP Redistribution

Each EIGRP routing process provides routing information to only one autonomous system. The Cisco IOS software must run a separate EIGRP process and maintain a separate routing database for each autonomous system that the software services. However, you can transfer routing information among routing databases.

In the following example, network 10.0.0.0 is configured under EIGRP autonomous system 1 and network 192.168.7.0 is configured under EIGRP autonomous system 101:

```
Router(config)# router eigrp 1
Router(config-router)# network 10.0.0.0
Router(config-router)# exit
Router(config)# router eigrp 101
Router(config-router)# network 192.168.7.0
```

In the following example, routes from the 192.168.7.0 network are redistributed into autonomous system 1 (without passing any other routing information from autonomous system 101):

```
Router(config)# access-list 3 permit 192.168.7.0
Router(config)# !
```

```
Router(config)# route-map 101-to-1 permit 10
Router(config-route-map)# match ip address 3
Router(config-route-map)# set metric 10000 100 1 255 1500
Router(config-route-map)# exit
Router(config)# router eigrp 1
Router(config-router)# redistribute eigrp 101 route-map 101-to-1
Router(config-router)#!
```

The following example is an alternative way to redistribute routes from the 192.168.7.0 network into autonomous system 1.This method does not allow you to set the metric for redistributed routes.

```
Router(config)# access-list 3 permit 192.168.7.0
Router(config)# !
Router(config)# router eigrp 1
Router(config-router)# redistribute eigrp 101
Router(config-router)# distribute-list 3 out eigrp 101
Router(config-router)# !
```

Example: Mutual Redistribution Between EIGRP and RIP

Consider a WAN at a university that uses the Routing Information Protocol (RIP) as an interior routing protocol. Assume that the university wants to connect its WAN to regional network 172.16.0.0, which uses the Enhanced Interior Gateway Routing Protocol (EIGRP) as the routing protocol. The goal in this case is to advertise the networks in the university network to devices in the regional network.

Mutual redistribution is configured between EIGRP and RIP in the following example:

```
Device(config)# access-list 10 permit 172.16.0.0
Device(config)# !
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# redistribute rip metric 10000 100 255 1 1500
Device(config-router)# default-metric 10
Device(config-router)# distribute-list 10 out rip
Device(config-router)# exit
Device(config)# router rip
Device(config-router)# redistribute eigrp 1
Device(config-router)# !
```

In this example, an EIGRP routing process is started. The **network** router configuration command specifies that network 172.16.0.0 (the regional network) is to send and receive EIGRP routing information. The **redistribute** router configuration command specifies that RIP-derived routing information be advertised in routing updates. The **default-metric** router configuration command assigns an EIGRP metric to all RIP-derived routes. The **distribute-list** router configuration command instructs the Cisco software to use access list 10 (not defined in this example) to limit the entries in each outgoing update. The access list prevents unauthorized advertising of university routes to the regional network.

Example: Mutual Redistribution Between EIGRP and BGP

In the following example, mutual redistribution is configured between the Enhanced Interior Gateway Routing Protocol (EIGRP) and the Border Gateway Protocol (BGP).

Routes from EIGRP routing process 101 are injected into BGP autonomous system 50000. A filter is configured to ensure that the correct routes are advertised, in this case, three networks. Routes from BGP autonomous system 50000 are injected into EIGRP routing process 101. The same filter is used.

```
Device(config)# ! All networks that should be advertised from R1 are controlled with ACLs:

Device(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.16.0.0 0.0.255.255
Device(config)# access-list 1 permit 172.25.0.0 0.0.255.255
Device(config)# ! Configuration for router R1:
Device(config)# router bgp 50000
Device(config-router)# network 172.18.0.0
Device(config-router)# network 172.16.0.0
Device(config-router)# neighbor 192.168.10.1 remote-as 2
Device(config-router)# neighbor 192.168.10.15 remote-as 1
Device(config-router)# neighbor 192.168.10.24 remote-as 3
Device(config-router)# redistribute eigrp 101
Device(config-router)# distribute-list 1 out eigrp 101
Device(config-router)# exit
Device(config)# router eigrp 101
Device(config-router)# network 172.25.0.0
Device(config-router)# redistribute bgp 50000
Device(config-router)# distribute-list 1 out bgp 50000
Device(config-router)# !
```

Caution BGP should be redistributed into an Interior Gateway Protocol (IGP) when there are no other suitable options. Redistribution from BGP into any IGP should be applied with proper filtering by using distribute lists, IP prefix lists, and route map statements to limit the number of prefixes.

Examples: OSPF Routing and Route Redistribution

OSPF typically requires coordination among many internal devices, area border routers (ABRs), and Autonomous System Boundary Routers (ASBRs). At a minimum, OSPF-based devices can be configured with all default parameter values, with no authentication, and with interfaces assigned to areas.

This section provides the following configuration examples:

- The first example shows simple configurations illustrating basic OSPF commands.
- The second example shows configurations for an internal device, ABR, and ASBR within a single, arbitrarily assigned OSPF autonomous system.
- The third example illustrates a more complex configuration and the application of various tools available for controlling OSPF-based routing environments.

Example: Basic OSPF Configurations

The following example shows a simple OSPF configuration that enables OSPF routing process 1, attaches Ethernet interface 0/0 to Area 0.0.0.0, and redistributes RIP into OSPF and OSPF into RIP:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# ip ospf cost 1
Router(config-if)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.17.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
```

```
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0.0.0.0
Router(config-router)# redistribute rip metric 1 subnets
Router(config-router)# exit
Router(config)# router rip
Router(config-router)# network 172.17.0.0
Router(config-router)# redistribute ospf 1
Router(config-router)# default-metric 1
Router(config-router)# !
```

The following example shows the assignment of four area IDs to four IP address ranges. In the example, OSPF routing process 1 is initialized, and four OSPF areas are defined: 10.9.50.0, 2, 3, and 0. Areas 10.9.50.0, 2, and 3 mask specific address ranges, whereas Area 0 enables OSPF for all other networks.

```
Router(config)# router ospf 1
Router(config-router)# network 172.18.20.0 0.0.0.255 area 10.9.50.0
Router(config-router)# network 172.18.0.0 0.0.255.255 area 2
Router(config-router)# network 172.19.10.0 0.0.0.255 area 3
Router(config-router)# network 0.0.0.0 255.255.255.255 area 0
Router(config-router)# exit
Router(config)# ! Ethernet interface 0/0 is in area 10.9.50.0:
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.18.20.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 1/0 is in area 2:
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.18.1.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 2/0 is in area 2:
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 172.18.2.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 3/0 is in area 3:
Router(config)# interface Ethernet 3/0
Router(config-if)# ip address 172.19.10.5 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 4/0 is in area 0:
Router(config)# interface Ethernet 4/0
Router(config-if)# ip address 172.19.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# ! Ethernet interface 5/0 is in area 0:
Router(config)# interface Ethernet 5/0
Router(config-if)# ip address 10.1.0.1 255.255.0.0
Router(config-if)# !
```

Each **network** router configuration command is evaluated sequentially, so the specific order of these commands in the configuration is important. The Cisco IOS software sequentially evaluates the *address wildcard-mask* pair for each interface. See the *IP Routing: Protocol-Independent Command Reference* for more information.

Consider the first **network** command. Area ID 10.9.50.0 is configured for the interface on which subnet 172.18.20.0 is located. Assume that a match is determined for Ethernet interface 0/0. Ethernet interface 0/0 is attached to Area 10.9.50.0 only.

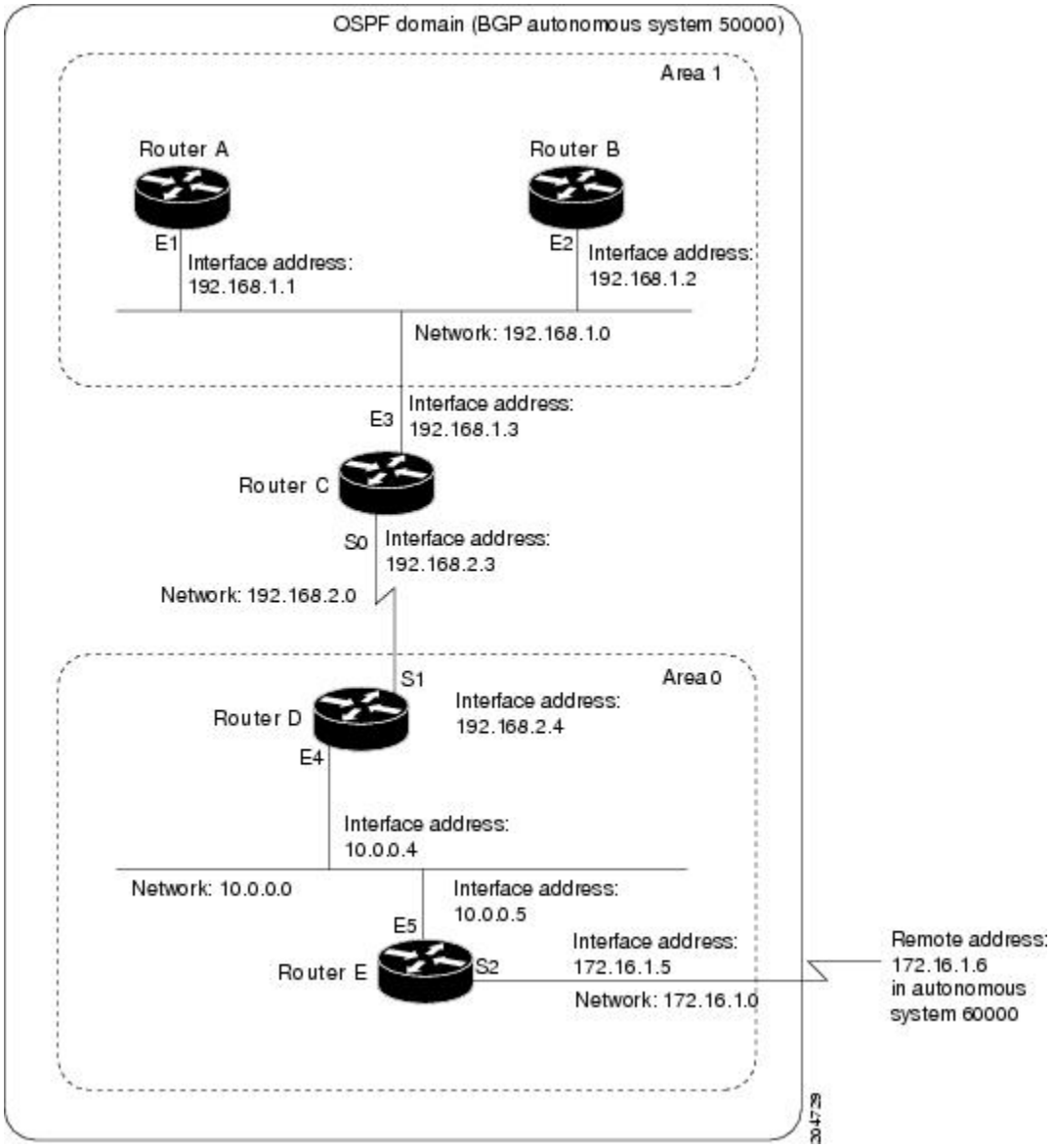
The second **network** command is evaluated next. For Area 2, all interfaces (except Ethernet interface 0/0) are evaluated. Assume that a match is determined for Ethernet interface 1/0. OSPF is then enabled for that interface, and Ethernet 1/0 is attached to Area 2.

This process of attaching interfaces to OSPF areas continues for all **network** commands. Note that the last **network** command in this example is a special case. With this command, all available interfaces (not explicitly attached to another area) are attached to Area 0.

Example: Internal Router ABR and ASBR Configurations

The figure below provides a general network map that illustrates a sample configuration for several routers within a single OSPF autonomous system.

Figure 5: Example OSPF Autonomous System Network Map



In this configuration, the following five routers are configured in OSPF autonomous system 1:

- Router A and Router B are both internal routers within area 1.
- Router C is an OSPF ABR. Note that for Router C, area 1 is assigned to E3 and Area 0 is assigned to S0.
- Router D is an internal router in Area 0 (backbone area). In this case, both **network** router configuration commands specify the same area (Area 0 or the backbone area).

- Router E is an OSPF ASBR. Note that BGP routes are redistributed into OSPF and that these routes are advertised by OSPF.

Note You don't have to include definitions of all areas in an OSPF autonomous system in the configuration of all routers in the autonomous system. You must define only the directly connected areas. In the example that follows, routes in Area 0 are learned by routers in area 1 (Router A and Router B) when the ABR (Router C) injects summary LSAs into Area 1.

Autonomous system 60000 is connected to the outside world via the BGP link to the external peer at IP address 172.16.1.6.

Here is an example configuration for the general network map shown in the figure above.

Router A Configuration—Internal Router

```
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# exit
```

Router B Configuration—Internal Router

```
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 192.168.1.2 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# exit
```

Router C Configuration—ABR

```
Router(config)# interface Ethernet 3/0
Router(config-if)# ip address 192.168.1.3 255.255.255.0
Router(config-if)# exit
Router(config)# interface Serial 0
Router(config-if)# ip address 192.168.2.3 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 1
Router(config-router)# network 192.168.2.0 0.0.0.255 area 0
Router(config-router)# exit
```

Router D Configuration—Internal Router

```
Router(config)# interface Ethernet 4/0
Router(config-if)# ip address 10.0.0.4 255.0.0.0
Router(config-if)# exit
Router(config)# interface Serial 1
Router(config-if)# ip address 192.168.2.4 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 192.168.2.0 0.0.0.255 area 0
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# exit
```

Router E Configuration—ASBR

```
Router(config)# interface Ethernet 5/0
Router(config-if)# ip address 10.0.0.5 255.0.0.0
Router(config-if)# exit
Router(config)# interface Serial 2
Router(config-if)# ip address 172.16.1.5 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 0
Router(config-router)# redistribute bgp 50000 metric 1 metric-type 1
Router(config-router)# exit
Router(config)# router bgp 50000
Router(config-router)# network 192.168.0.0
Router(config-router)# network 10.0.0.0
Router(config-router)# neighbor 172.16.1.6 remote-as 60000
```

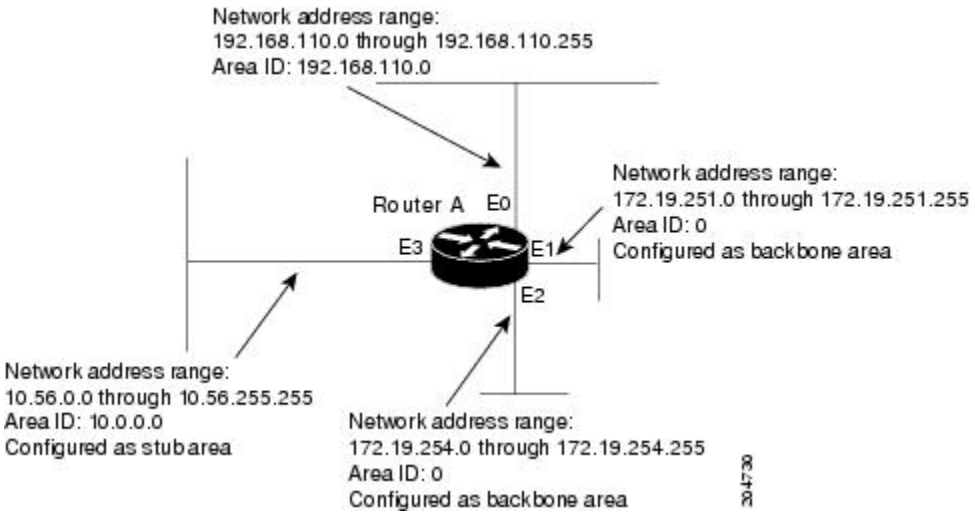
Example: Complex OSPF Configuration

The following example configuration accomplishes several tasks in setting up an ABR. These tasks can be split into the following two general categories:

- Basic OSPF configuration
- Route redistribution

The figure below illustrates the network address ranges and area assignments for interfaces.

Figure 6: Interface and Area Specifications for the OSPF Configuration



The basic configuration tasks in this example are as follows:

- Configure address ranges for Ethernet interface 0 through Ethernet interface 3.
- Enable OSPF on each interface.
- Set up an OSPF authentication password for each area and network.
- Assign link-state metrics and other OSPF interface configuration options.

- Create a *stub area* with area ID 10.0.0.0. (Note that the **authentication** and **stub** options of the **area** router configuration command are specified with separate **area** command entries, but they can be merged into a single **area** command.)
- Specify the backbone area (Area 0).

Configuration tasks associated with route redistribution are as follows:

- Redistribute EIGRP and RIP into OSPF with various options set (including **metric-type**, **metric**, **tag**, and **subnet**).
- Redistribute EIGRP and OSPF into RIP.

The following is a sample OSPF configuration:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 192.168.110.201 255.255.255.0
Router(config-if)# ip ospf authentication-key abcdefgh
Router(config-if)# ip ospf cost 10
Router(config-if)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.19.251.201 255.255.255.0
Router(config-if)# ip ospf authentication-key ijklmnop
Router(config-if)# ip ospf cost 20
Router(config-if)# ip ospf retransmit-interval 10
Router(config-if)# ip ospf transmit-delay 2
Router(config-if)# ip ospf priority 4
Router(config-if)# exit
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 172.19.254.201 255.255.255.0
Router(config-if)# ip ospf authentication-key abcdefgh
Router(config-if)# ip ospf cost 10
Router(config-if)# exit
Router(config)# interface Ethernet 3/0
Router(config-if)# ip address 10.56.0.201 255.255.0.0
Router(config-if)# ip ospf authentication-key ijklmnop
Router(config-if)# ip ospf cost 20
Router(config-if)# ip ospf dead-interval 80
Router(config-if)# exit
```

In the following configuration, OSPF is on network 172.19.0.0:

```
Router(config)# router ospf 1
Router(config-router)# network 10.0.0.0 0.255.255.255 area 10.0.0.0
Router(config-router)# network 192.168.110.0 0.0.0.255 area 192.68.110.0
Router(config-router)# network 172.19.0.0 0.0.255.255 area 0
Router(config-router)# area 0 authentication
Router(config-router)# area 10.0.0.0 stub
Router(config-router)# area 10.0.0.0 authentication
Router(config-router)# area 10.0.0.0 default-cost 20
Router(config-router)# area 192.168.110.0 authentication
Router(config-router)# area 10.0.0.0 range 10.0.0.0 255.0.0.0
Router(config-router)# area 192.168.110.0 range 192.168.110.0 255.255.255.0
Router(config-router)# area 0 range 172.19.251.0 255.255.255.0
Router(config-router)# area 0 range 172.19.254.0 255.255.255.0
Router(config-router)# redistribute eigrp 200 metric-type 2 metric 1 tag 200 subnets
Router(config-router)# redistribute rip metric-type 2 metric 1 tag 200
Router(config-router)# exit
```

In the following configuration, EIGRP autonomous system 1 is on 172.19.0.0:

```
Router(config)# router eigrp 1
Router(config-router)# network 172.19.0.0
Router(config-router)# exit
Router(config)# ! RIP for 192.168.110.0:
Router(config)# router rip
Router(config-router)# network 192.168.110.0
Router(config-router)# redistribute eigrp 1 metric 1
Router(config-router)# redistribute ospf 201 metric 1
Router(config-router)# exit
```

Example: Default Metric Values Redistribution

The following example shows how a router in autonomous system 1 is configured to run both RIP and EIGRP. The example advertises EIGRP-derived routes using RIP and assigns the EIGRP-derived routes a RIP metric of 10.

```
Router(config)# router rip
Router(config-router)# default-metric 10
Router(config-router)# redistribute eigrp 1
Router(config-router)# exit
```

Example: Route Map

The examples in this section illustrate the use of redistribution with and without route maps. Examples from both the IP and Connectionless Network Service (CLNS) routing protocols are given below. The following example shows how to redistribute all OSPF routes into EIGRP:

```
Router(config)# router eigrp 1
Router(config-router)# redistribute ospf 101
Router(config-router)# exit
```

The following example shows how to redistribute RIP routes, with a hop count equal to 1, into OSPF. These routes will be redistributed into OSPF as external LSAs with a metric of 5, metric-type of type 1, and a tag equal to 1.

```
Router(config)# router ospf 1
Router(config-router)# redistribute rip route-map rip-to-ospf
Router(config-router)# exit
Router(config)# route-map rip-to-ospf permit
Router(config-route-map)# match metric 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type 1
Router(config-route-map)# set tag 1
Router(config-route-map)# exit
```

The following example shows how to redistribute OSPF learned routes with tag 7 as a RIP metric of 15:

```
Router(config)# router rip
Router(config-router)# redistribute ospf 1 route-map 5
Router(config-router)# exit
Router(config)# route-map 5 permit
Router(config-route-map)# match tag 7
Router(config-route-map)# set metric 15
```

The following example shows how to redistribute OSPF intra-area and inter-area routes with next-hop routers on serial interface 0/0 into BGP with an INTER_AS metric of 5:

```
Router(config)# router bgp 50000
Router(config-router)# redistribute ospf 1 route-map 10
Router(config-router)# exit
Router(config)# route-map 10 permit
Router(config-route-map)# match route-type internal
Router(config-route-map)# match interface serial 0
Router(config-route-map)# set metric 5
```

The following example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first type is OSPF external IP routes with tag 5; these routes are inserted into Level 2 IS-IS link-state packets (LSPs) with a metric of 5. The second type is ISO-IGRP derived CLNS prefix routes that match CLNS access list 2000; these routes will be redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
Router(config)# router isis
Router(config-router)# redistribute ospf 1 route-map 2
Router(config-router)# redistribute iso-igrp nsfnet route-map 3

Router(config-router)# exit
Router(config)# route-map 2 permit
Router(config-route-map)# match route-type external
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# route-map 3 permit
Router(config-route-map)# match address 2000
Router(config-route-map)# set metric 30
Router(config-route-map)# exit
```

With the following configuration, OSPF external routes with tags 1, 2, 3, and 5 are redistributed into RIP with metrics of 1, 1, 5, and 5, respectively. The OSPF routes with a tag of 4 are not redistributed.

```
Router(config)# router rip
Router(config-router)# redistribute ospf 101 route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 1 2
Router(config-route-map)# set metric 1
Router(config-route-map)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match tag 3
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
Router(config)# route-map 1 deny
Router(config-route-map)# match tag 4
Router(config-route-map)# exit
Router(config)# route map 1 permit
Router(config-route-map)# match tag 5
Router(config-route-map)# set metric 5
Router(config-route-map)# exit
```

Given the following configuration, a RIP learned route for network 172.18.0.0 and an ISO-IGRP learned route with prefix 49.0001.0002 will be redistributed into an IS-IS Level 2 LSP with a metric of 5:

```
Router(config)# router isis
```

```
Router(config-router)# redistribute rip route-map 1
Router(config-router)# redistribute iso-igrp remote route-map 1
Router(config-router)# exit
Router(config)# route-map 1 permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# match clns address 2
Router(config-route-map)# set metric 5
Router(config-route-map)# set level level-2
Router(config-route-map)# exit
Router(config)# access-list 1 permit 172.18.0.0 0.0.255.255
Router(config)# clns filter-set 2 permit 49.0001.0002...
```

The following configuration example illustrates how a route map is referenced by the **default-information** router configuration command. This type of reference is called conditional default origination. OSPF will originate the default route (network 0.0.0.0) with a type 2 metric of 5 if 172.20.0.0 is in the routing table.

```
Router(config)# route-map ospf-default permit
Router(config-route-map)# match ip address 1
Router(config-route-map)# set metric 5
Router(config-route-map)# set metric-type type-2
Router(config-route-map)# exit
Router(config)# access-list 1 172.20.0.0 0.0.255.255
Router(config)# router ospf 101
Router(config-router)# default-information originate route-map ospf-default
```

Example: Passive Interface

In OSPF, hello packets are not sent on an interface that is specified as passive. Hence, the router will not be able to discover any neighbors, and none of the OSPF neighbors will be able to see the router on that network. In effect, this interface will appear as a stub network to the OSPF domain. This configuration is useful if you want to import routes associated with a connected network into the OSPF domain without any OSPF activity on that interface.

The **passive-interface** router configuration command is typically used when the wildcard specification on the **network** router configuration command configures more interfaces than is desirable. The following configuration causes OSPF to run on all subnets of 172.18.0.0:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.18.1.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 172.18.2.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface Ethernet 2/0
Router(config-if)# ip address 172.18.3.1 255.255.255.0
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0

Router(config-router)# exit
```

If you do not want OSPF to run on 172.18.3.0, enter the following commands:

```
Router(config)# router ospf 1
Router(config-router)# network 172.18.0.0 0.0.255.255 area 0
Router(config-router)# passive-interface Ethernet 2
Router(config-router)# exit
```

Example: Configuring Default Passive Interfaces

The following example shows how to configure network interfaces, set all interfaces that are running OSPF as passive, and then enable serial interface 0/0:

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip address 172.19.64.38 255.255.255.0 secondary
Router(config-if)# ip address 172.19.232.70 255.255.255.240
Router(config-if)# no ip directed-broadcast
Router(config-if)# exit
Router(config)# interface Serial 0/0
Router(config-if)# ip address 172.24.101.14 255.255.255.252
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# exit
Router(config)# interface TokenRing 0
Router(config-if)# ip address 172.20.10.4 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# ring-speed 16
Router(config-if)# exit
Router(config)# router ospf 1
Router(config-router)# passive-interface default
Router(config-router)# no passive-interface Serial 0/0
Router(config-router)# network 172.16.10.0 0.0.0.255 area 0
Router(config-router)# network 172.19.232.0 0.0.0.255 area 4
Router(config-router)# network 172.24.101.0 0.0.0.255 area 4
Router(config-router)# exit
```

Example: Policy-Based Routing

The following example provides two sources with equal access to two different service providers. Packets that arrive on asynchronous interface 1/0/0 from the source 10.1.1.1 are sent to the device at 172.16.6.6 if the device has no explicit route for the destination of the packet. Packets that arrive from the source 172.17.2.2 are sent to the device at 192.168.7.7 if the device has no explicit route for the destination of the packet. All other packets for which the device has no explicit route to the destination are discarded.

```
Device(config)# access-list 1 permit ip 10.1.1.1
Device(config)# access-list 2 permit ip 172.17.2.2
Device(config)# interface async 1/0/0
Device(config-if)# ip policy route-map equal-access
Device(config-if)# exit
Device(config)# route-map equal-access permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip default next-hop 172.16.6.6
Device(config-route-map)# exit
Device(config)# route-map equal-access permit 20
Device(config-route-map)# match ip address 2
Device(config-route-map)# set ip default next-hop 192.168.7.7
Device(config-route-map)# exit
Device(config)# route-map equal-access permit 30
Device(config-route-map)# set default interface null 0
Device(config-route-map)# exit
```

Example: Policy Routing with Cisco Express Forwarding

The following example shows how to configure policy routing with Cisco Express Forwarding. The route is configured to verify that the next hop 10.0.0.8 of the route map named test is a Cisco Discovery Protocol neighbor before the device tries to policy-route to it.

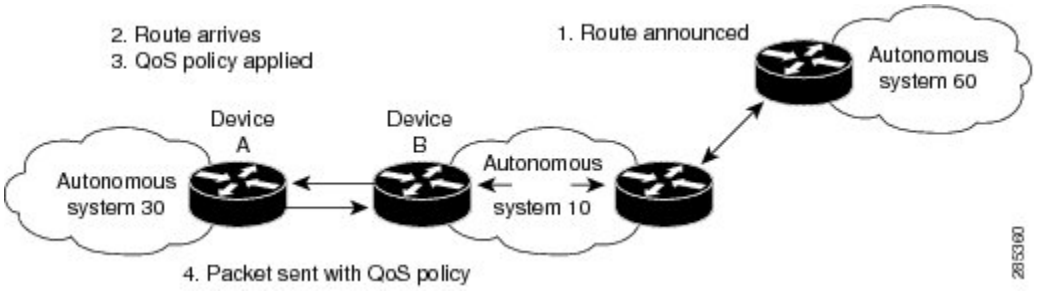
```
Device(config)# ip cef
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip route-cache flow
Device(config-if)# ip policy route-map test
Device(config-if)# exit
Device(config)# route-map test permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip precedence priority
Device(config-route-map)# set ip next-hop 10.0.0.8
Device(config-route-map)# set ip next-hop verify-availability
Device(config-route-map)# exit
Device(config)# route-map test permit 20
Device(config-route-map)# match ip address 101
Device(config-route-map)# set interface Ethernet 0/0/3
Device(config-route-map)# set ip tos max-throughput
Device(config-route-map)# exit
```

Example: Configuring QoS Policy Propagation via BGP

The following example shows how to create route maps to match access lists, Border Gateway Protocol (BGP) community lists, and BGP autonomous system paths, and apply IP precedence to routes learned from neighbors.

In the figure below, Device A learns routes from autonomous system 10 and autonomous system 60. The quality of service (QoS) policy is applied to all packets that match defined route maps. Any packets from Device A to autonomous system 10 or autonomous system 60 are sent the appropriate QoS policy, as the numbered steps in the figure indicate.

Figure 7: Device Learning Routes and Applying QoS Policy



Device A Configuration

```
interface serial 5/0/0/1:0
ip address 10.28.38.2 255.255.255.0
bgp-policy destination ip-prec-map
no ip mroute-cache
no cdp enable
frame-relay interface-dlci 20 IETF
router bgp 30
```

```
 table-map precedence-map
 neighbor 10.20.20.1 remote-as 10
 neighbor 10.20.20.1 send-community
!
ip bgp-community new-format
!
! Match community 1 and set the IP precedence to priority
route-map precedence-map permit 10
 match community 1
 set ip precedence priority
!
! Match community 2 and set the IP precedence to immediate
route-map precedence-map permit 20
 match community 2
 set ip precedence immediate
!
! Match community 3 and set the IP precedence to flash
route-map precedence-map permit 30
 match community 3
 set ip precedence flash
!
! Match community 4 and set the IP precedence to flash-override
route-map precedence-map permit 40
 match community 4
 set ip precedence flash-override
!
! Match community 5 and set the IP precedence to critical
route-map precedence-map permit 50
 match community 5
 set ip precedence critical
!
! Match community 6 and set the IP precedence to internet
route-map precedence-map permit 60
 match community 6
 set ip precedence internet
!
! Match community 7 and set the IP precedence to network
route-map precedence-map permit 70
 match community 7
 set ip precedence network
!
! Match ip address access list 69 or match autonomous system path 1
! and set the IP precedence to critical
route-map precedence-map permit 75
 match ip address 69
 match as-path 1
 set ip precedence critical
!
! For everything else, set the IP precedence to routine
route-map precedence-map permit 80
 set ip precedence routine
!
! Define community lists
ip community-list 1 permit 60:1
ip community-list 2 permit 60:2
ip community-list 3 permit 60:3
ip community-list 4 permit 60:4
ip community-list 5 permit 60:5
ip community-list 6 permit 60:6
ip community-list 7 permit 60:7
!
! Define the AS path
ip as-path access-list 1 permit ^10_60
!
```

```
! Define the access list
access-list 69 permit 10.69.0.0
```

Device B Configuration

```
router bgp 10
 neighbor 10.30.30.1 remote-as 30
 neighbor 10.30.30.1 send-community
 neighbor 10.30.30.1 route-map send_community out
!
ip bgp-community new-format
!
! Match prefix 10 and set community to 60:1
route-map send_community permit 10
 match ip address 10
 set community 60:1
!
! Match prefix 20 and set community to 60:2
route-map send_community permit 20
 match ip address 20
 set community 60:2
!
! Match prefix 30 and set community to 60:3
route-map send_community permit 30
 match ip address 30
 set community 60:3
!
! Match prefix 40 and set community to 60:4
route-map send_community permit 40
 match ip address 40
 set community 60:4
!
! Match prefix 50 and set community to 60:5
route-map send_community permit 50
 match ip address 50
 set community 60:5
!
! Match prefix 60 and set community to 60:6
route-map send_community permit 60
 match ip address 60
 set community 60:6
!
! Match prefix 70 and set community to 60:7
route-map send_community permit 70
 match ip address 70
 set community 60:7
!
! For all others, set community to 60:8
route-map send_community permit 80
 set community 60:8
!
! Define access lists
access-list 10 permit 10.61.0.0
access-list 20 permit 10.62.0.0
access-list 30 permit 10.63.0.0
access-list 40 permit 10.64.0.0
access-list 50 permit 10.65.0.0
access-list 60 permit 10.66.0.0
access-list 70 permit 10.67.0.0
```

Example: Managing Authentication Keys

The following example shows how to configure a key chain named kc1. In this example, the software will always accept and send ks1 as a valid key. The key ks2 will be accepted from 1:30 p.m. to 3:30 p.m. and be sent from 2:00 p.m. to 3:00 p.m. The overlap allows for migration of keys or discrepancy in the set time of the router.

```
Router(config)# interface Ethernet 0/0
Router(config-if)# ip rip authentication key-chain kc1
Router(config-if)# ip rip authentication mode md5
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 172.19.0.0
Router(config-router)# version 2
Router(config-router)# exit
Router(config)# key chain kc1
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string ks1
Router(config-keychain-key)# key 2
Router(config-keychain-key)# key-string ks2
Router(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 2005 duration 7200
Router(config-keychain-key)# send-lifetime 14:00:00 Jan 25 2005 duration 3600
Router(config-keychain-key)# key 3
Router(config-keychain-key)# key-string ks3
Router(config-keychain-key)# accept-lifetime 14:30:00 Jan 25 2005 duration 7200
Router(config-keychain-key)# send-lifetime 15:00:00 Jan 25 2005 duration 3600
Router(config-keychain-key)# exit
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

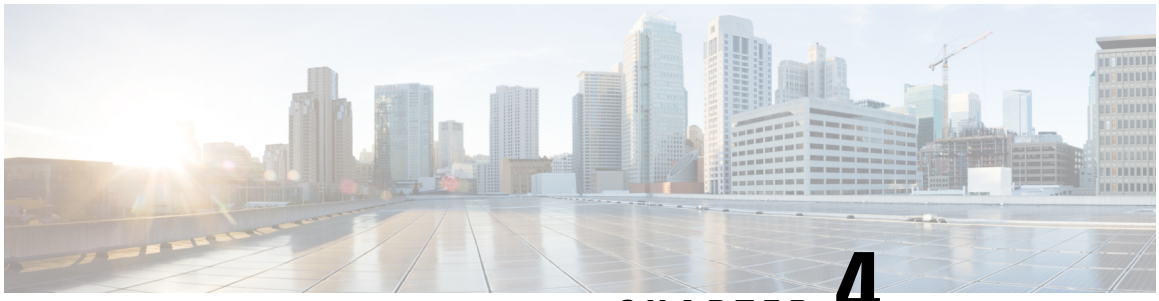
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring IP Routing Protocol-Independent Features

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for Configuring IP Routing Protocol-Independent Features



CHAPTER 4

Configuring Route Leaking and Redistribution

This chapter contains the following sections:

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Route Leaking and Redistribution

Overview of Route Leaking and Redistribution

Route leaking between the global or default VRF (transport VPN) and service VPNs allows you to share common services that multiple VPNs need to access. With this feature, routes are replicated through bidirectional route leaking between the global VRF (also known as transport VPN) and service VPNs. Route leaking between VRFs is done using Routing Information Base (RIB).

Note In the context of Cisco SD-WAN, the terms VRF and VPN are used interchangeably. Although Cisco IOS XE SD-WAN devices use VRFs for segmentation and network isolation, the VPN feature template is used to configure them using Cisco vManage. When you use Cisco vManage to configure VPNs for Cisco IOS XE SD-WAN devices, Cisco vManage automatically converts the VPN configuration to VRF configuration.

To apply the leaked routes to the routing neighbors, you can redistribute the leaked routes between the global VRF and service VPNs. In addition to running multiple routing protocols simultaneously, you can redistribute routes from one routing protocol to another.

Features of Route Leaking and Redistribution

- Leak routes between the global VRF and service VPNs directly.
- Leak multiple service VPNs to the global VRF.
- Apply different route policies using route-map during route replication and redistribution.
- Use route-maps to filter routes using match operations before leaking them.
- Configure these features using both—Cisco vManage and CLI.
- When routes are leaked and redistributed between the global VRF and service VPNs, route properties such as metric, source VPN information, tags, administrative distance, and route origin are retained and carried to the destination protocol.

Typical Use Cases and Benefits

- **Service Provider Central Services:** SP Central services under MPLS can be directly accessed without having to duplicate them for each VPN. This makes accessing central services easier and more efficient.
- **Migration:** With route leaking, branches that have migrated to Cisco SD-WAN can directly access non-migrated branches bypassing the hub, thus providing improved application SLAs.
- **Centralized Network Management:** You can manage the control plane and service-side equipment through the underlay.

How Route Preference is Determined

If a route is replicated or leaked between the global VRF and service VPNs, the following rule determines the route preference.

For a device that receives a route from two sources in which both these routes use the same source VRFs, if one of these routes is replicated, then the non-replicated route is preferred.

If the aforementioned rule doesn't apply, the following rules determine the order of route preference:

1. Prefer the route with smaller administrative distance.
2. Prefer the route with smaller default administrative distance.
3. Prefer a non-replicated route over a replicated route.
4. Compare original VRF-names. Prefer the route with the lexicographically smaller VRF-name.
5. Compare original subaddress families. Prefer unicast routing over multicast routing.
6. Prefer the oldest route.

Supported Protocols

The following protocols are supported for route leaking between the global VRF and service VPNs.

- Connected
- Static
- BGP
- OSPF
- EIGRP

The following protocols are the supported destination and source protocols for route redistribution between the service VPNs and global VRF.

Source Protocols

- Connected
- Static
- BGP
- OSPF
- EIGRP

Destination Protocols

- BGP
- OSPF
- EIGRP

Note The EIGRP protocol can be used only on service VPNs and not on the global VRF. Therefore, route leaking is supported only for routes from the global VRF to service VPNs.

Restrictions for Route Leaking and Redistribution

- The EIGRP protocol can be used only on service VPNs and not on the global VRF. Therefore, route leaking isn't supported for routes from service VPNs to the global VRF and between service VPNs for the EIGRP protocol.
- Service-side NAT isn’t supported with route leaking between the global VRF and service VPNs.
- This feature is specific to leaking routes between the global VRF and service VPNs only and doesn't support route leaking from a service VPN to another service VPN.
- IPv6 address family isn’t supported.
- Only prefix-lists, tags, communities, and metrics can be matched in route maps that are used to filter leaked routes.

- While configuring route leaking for a VRF, the **route-replicate** command under the **global-address-family ipv4** command shouldn't have the keyword **all** specified as the protocol for the unicast option to prevent route looping.

```
global-address-family ipv4
  route-replicate from vrf <vrf> unicast all
```

- In this example, the keyword all should be replaced with specific protocol name as shown here:

```
global-address-family ipv4
  route-replicate from vrf <vrf> unicast connected
```

How to Configure Route Leaking and Redistribution

You can leak routes in the global routing table (global VRF) or service VPN, and then redistribute these routes. The following scenarios are supported:

- Leaking and redistribution of routes from a service VPN into the global VRF
- Leaking and redistribution of routes from the global VRF into a service VPN

Configuring Route Leaking and Redistribution from Service VPN into Global VRF

The following procedure shows how to leak and redistribute routes from a service VPN into the global VRF in the BGP protocol.

```
configure terminal
 global-address-family ipv4 unicast
  route-replicate from vrf src-vrf-name unicast src_protocol [src_protocol_id] [route-map
route-map-name]

 router bgp router_instance_id
  address-family ipv4
   redistribute vrf src-vrf-name src_protocol [src_protocol_id] [route-map route-map-name]
```

Note Use the **router ospf** command instead of the **router bgp** command to configure a routing process for the OSPF routing protocol.

The following procedure shows how to leak and redistribute routes from a service VPN into the global VRF in the EIGRP protocol.

```
configure terminal
 global-address-family ipv4 unicast
  route-replicate from vrf src-vrf-name unicast src_protocol [src_protocol_id] [route-map
route-map-name]

 router eigrp autonomous-system-number
  address-family ipv4
   redistribute vrf src-vrf-name src_protocol [src_protocol_id] [metric bandwidth-metric
```

```
delay-metric reliability-metric effective-bandwidth-metric mtu-bytes] [route-map
route-map-name]
```

Note The *src_protocol_id* is optional because static and connected routes do not have the instance IDs. However, BGP, OSPF, and EIGRP have these instance IDs.

Configuring Route Leaking and Redistribution from Global VRF into Service VPN

The following procedure shows how to leak and redistribute routes from the global VRF into a service VPN in the BGP protocol.

```
configure terminal
 vrf definition vrf_name
  address-family ipv4
   route-replicate from vrf global unicast src_protocol [src_protocol_id] [route-map
route-map-name]

 router bgp router_instance_id
  address-family ipv4 vrf vrf_name
   redistribute vrf global src_protocol [src_protocol_id] [route-map route-map-name]
```

The following procedure shows how to leak and redistribute routes from the global VRF into a service VPN in the OSPF protocol.

```
configure terminal
 vrf definition vrf_name
  address-family ipv4
   route-replicate from vrf global unicast src_protocol [src_protocol_id] [route-map
route-map-name]

 router ospf router_instance_id vrf vrf_name
   redistribute vrf global src_protocol [src_protocol_id] [route-map route-map-name]
```

The following procedure shows how to leak and redistribute routes from the global VRF into a service VPN in the EIGRP protocol.

```
configure terminal
 vrf definition vrf_name
  address-family ipv4
   route-replicate from vrf global unicast src_protocol [src_protocol_id] [route-map
route-map-name]

 router eigrp autonomous-system-number
  address-family ipv4 vrf vrf_name
   redistribute vrf global src_protocol [src_protocol_id] [metric bandwidth-metric
delay-metric reliability-metric effective-bandwidth-metric mtu-bytes] [route-map
route-map-name]
```

Note The *src_protocol_id* is optional because static and connected routes do not have the instance IDs. However, BGP, OSPF, and EIGRP have these instance IDs.

Examples: Configure Route Leaking and Redistribution

Example: Leak Routes between Global VRF and Service VPNs

These examples show how to configure route leaking between a global VRF and a service VPN. In this example, VRF 103 is the service VPN. This example shows that the connected routes are leaked into VRF 103 from the global VRF, similarly, the same connected routes are leaked from VRF 103 to the global VRF.

```
vrf definition 103
 !
  address-family ipv4
   route-replicate from vrf global unicast connected
!
global-address-family ipv4
  route-replicate from vrf 103 unicast connected
  exit-address-family
```

Verify Configuration

The following examples shows how to view the leaked routes.

Note In the output, leaked routes are represented by a **+** sign next to the route leaked. Example: C+ denotes that a connected route was leaked.

```
Device#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
& - replicated local route overrides by connected

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 14 subnets, 2 masks
O 10.1.14.0/24 [110/11] via 10.1.15.13, 00:02:22, GigabitEthernet1
C 10.1.15.0/24 is directly connected, GigabitEthernet1
L 10.1.15.15/32 is directly connected, GigabitEthernet1
O 10.1.16.0/24 [110/11] via 10.1.15.13, 00:02:22, GigabitEthernet1
C 10.1.17.0/24 is directly connected, GigabitEthernet2
L 10.1.17.15/32 is directly connected, GigabitEthernet2
172.16.0.0/12 is subnetted, 1 subnets
[170/10880] via 192.168.24.17(103), 01:04:13, GigabitEthernet5.103
192.168.0.0/16 is variably subnetted, 2 subnets, 2 masks
C + 192.0.2.0/24  is directly connected, GigabitEthernet5.103
L & 192.168.24.15/16 is directly connected, GigabitEthernet5.103
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 203.0.113.0/24 is directly connected, GigabitEthernet6
L 203.0.113.15/32 is directly connected, GigabitEthernet6
10.20.0.0/8 is variably subnetted, 2 subnets, 2 masks
```

```
C 198.51.100.0/24 is directly connected, GigabitEthernet7
L 198.51.100.15/24 is directly connected, GigabitEthernet7
192.0.2.0/32 is subnetted, 1 subnets
O E2 100.100.100.100 [110/20] via 10.1.15.13, 00:02:22, GigabitEthernet1
172.16.0.0/32 is subnetted, 1 subnets
O E2 172.16.255.14 [110/20] via 10.1.15.13, 00:02:22, GigabitEthernet1
```

View Routes Leaked From Global VRF to Service VPN

Use the **show ip route vrf** <*vrf id*> command to view the routes leaked from the global VRF to the service VPN.

Note In the output, leaked routes are denoted by a **+** sign next to the route leaked. Example: C+ denotes that a connected route was leaked.

```
Device#show ip route vrf 103
Routing Table: 103
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
& - replicated local route overrides by connected

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 14 subnets, 2 masks
C + 10.0.1.0/24 is directly connected, GigabitEthernet9
L & 10.0.1.15/32 is directly connected, GigabitEthernet9
C + 10.0.20.0/24 is directly connected, GigabitEthernet4
L & 10.0.20.15/32 is directly connected, GigabitEthernet4
C + 10.0.100.0/24 is directly connected, GigabitEthernet8
L & 10.0.100.15/32 is directly connected, GigabitEthernet8
C + 10.1.15.0/24 is directly connected, GigabitEthernet1
L & 10.1.15.15/32 is directly connected, GigabitEthernet1
C + 10.1.17.0/24 is directly connected, GigabitEthernet2
L & 10.1.17.15/32 is directly connected, GigabitEthernet2
172.16.0.0/12 is subnetted, 1 subnets
D EX 172.16.20.20
[170/10880] via 192.168.24.17, 01:04:07, GigabitEthernet5.103
192.168.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 192.0.2.0/24 is directly connected, GigabitEthernet5.103
L 192.168.24.15/16 is directly connected, GigabitEthernet5.103
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C + 203.0.113.0/24 is directly connected, GigabitEthernet6
L & 203.0.113.15/32 is directly connected, GigabitEthernet6
10.20.0.0/8 is variably subnetted, 2 subnets, 2 masks
C + 198.51.100.0/24 is directly connected, GigabitEthernet7
L & 198.51.100.15/24 is directly connected, GigabitEthernet7
192.0.2.0/32 is subnetted, 1 subnets
```

Example: Filter Routes Before Leaking

To filter the routes leaked between the global VRF and the service VRF, you can apply a route map as shown in this example.

```
vrf definition 103
 !
  address-family ipv4
   route-replicate from vrf global unicast connected route-map myRouteMap permit 10
    match ip address prefix-list pList seq 5 permit 10.1.17.0/24
!
```

Verify Configuration

Note In this output, leaked routes are denoted by a **+** sign next to the route leaked. Example: C+ denotes that a connected route was leaked.

```
Device#show ip route vrf 103

Routing Table: 1
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
& - replicated local route overrides by connected

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C + 10.1.17.0/24 is directly connected, GigabitEthernet2
L & 10.1.17.15/32 is directly connected, GigabitEthernet2
m 10.1.18.0/24 [251/0] via 172.16.255.14, 19:01:28, Sdwan-system-intf
m 10.2.2.0/24 [251/0] via 172.16.255.11, 17:28:44, Sdwan-system-intf
m 10.2.3.0/24 [251/0] via 172.16.255.11, 17:26:50, Sdwan-system-intf
C 10.20.24.0/24 is directly connected, GigabitEthernet5
L 10.20.24.15/32 is directly connected, GigabitEthernet5
m 10.20.25.0/24 [251/0] via 172.16.255.11, 16:14:18, Sdwan-system-intf
172.16.0.0/32 is subnetted, 3 subnets
m 172.16.255.112 [251/0] via 172.16.255.11, 17:28:44, Sdwan-system-intf
O E2 172.16.255.117 [110/20] via 10.20.24.17, 1d11h, GigabitEthernet5
m 172.16.255.118 [251/0] via 172.16.255.11, 16:14:18, Sdwan-system-intf
```

To monitor leaked routes, use the **show ip cef** command. The output shows replicated or leaked routes.

```
Device#show ip cef 10.1.17.0 internal
10.1.17.0/24, epoch 2, flags [rcv], refcnt 6, per-destination sharing
[connected cover 10.1.17.0/24 replicated from 1]
sources: I/F
feature space:
Broker: linked, distributed at 4th priority
subblocks:
gsb Connected receive chain(0): 0x7F6B4315DB80
Interface source: GigabitEthernet5 flags: none flags3: none
Dependent covered prefix type cover need deagg, cover 10.20.24.0/24
```

```
ifnums: (none)
path list 7F6B47831168, 9 locks, per-destination, flags 0x41 [shble, hwcn]
path 7F6B3D9E7B70, share 1/1, type receive, for IPv4
receive for GigabitEthernet5
output chain:
receive
```

Example: Redistribute BGP Route into OSPF and EIGRP Protocols

This example shows how to replicate BGP routes from the global VRF into a service VPN.

```
Device(config)# vrf definition 2
Device(config-vrf)# address-family ipv4
Device(config-ipv4)# route-replicate from vrf global unicast bgp 1
Router(config-ipv4)# commit
```

This example shows how redistribute BGP Routes in the global VRF to EIGRP in the service VPN.

Note The redistribution of BGP routes into other protocols is supported only if the `bgp redistribute-internal` configuration is present in the BGP route.

```
Device(config)# router eigrp test
Device(config-router)# address-family ipv4 unicast vrf 2 autonomous-system 100
Device(config-router-af)# topology base
Device(config-router-af-topology)# redistribute vrf global bgp 1 metric 10000 100 200 1
1500
Device(config-ipv4)# commit

* Here we are redistributing BGP routes in global VRF to EIGRP in VRF 2.
* Routes replication must be done before doing inter VRF redistribution.
------------------------------------------------------------------------------
```

Verify Configuration

View BGP Route is not Present in Global VRF Before Configuring

Use the **show ip route bgp** command to view whether the BGP route is present in the global VRF before configuring.

```
Device#show ip route bgp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
& - replicated local route overrides by connected

Gateway of last resort is not set

10.0.0.0/9 is subnetted, 1 subnets
B 172.16.255.1 [200/20] via 10.1.15.14, 00:00:25
Device#
```

```
* We have a BGP route in the global VRF.
```

View BGP Route is not Present in Service VPN Before Configuring

Use the **show ip route vrf** <*vrf id*> [**protocol**] command to view the BGP route in the service VPN.

```
Device#show ip route vrf 2 bgp

Routing Table: 2
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
& - replicated local route overrides by connected

Gateway of last resort is not set

Device#

* We do not have any BGP route in VRF 2.
```

View BGP Route After Configuring

Use the **show running config** [*configuration-hierarchy*] | **details** command to verify if the replication configuration exists.

```
Device#show running-config | section vrf definition 2
vrf definition 2
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv4
  route-replicate from vrf global unicast bgp 1
 exit-address-family
Device#

* We have successfully applied the route-replicate configuration.
* In our example we are replicating bgp 1 routes from global VRF to VRF 2.
------------------------------------------------------------------------------
```

View BGP Route From Global VRF is Replicated into Service VPN After Configuring

Use the **show ip route vrf** <*vrf id*> [**protocol**] command to view the BGP route in the service VPN.

```
Device#show ip route vrf 2 bgp

Routing Table: 2
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
H - NHRP, G - NHRP registered, g - NHRP registration summary
o - ODR, P - periodic downloaded static route, l - LISP
```

```
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
& - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/9 is subnetted, 1 subnets
B   +    172.16.255.1 [200/20] via 10.1.15.14, 00:04:01
Device#

* After route replication, we can see that the BGP route in the global VRF has been replicated
 into VRF 2.
* + sign indicates replicated routes.
------------------------------------------------------------------------------
```

View EIGRP Configuration Without BGP Redistribution Information

```
Device#show running-config | section router eigrp
router eigrp test
 !
 address-family ipv4 unicast vrf 2 autonomous-system 100
  !
  topology base
  exit-af-topology
  network 10.0.0.0
 exit-address-family
Router#
```

View EIGRP Topology Table

Use the **show eigrp address-family ipv4 vrf**<*vrf-num*>**topology** command to view the BGP route in the service VRF table.

```
Device#show eigrp address-family ipv4 vrf 2 topology
EIGRP-IPv4 VR(test) Topology Table for AS(100)/ID(10.10.10.2)
           Topology(base) TID(0) VRF(2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.0.0.0/8, 1 successors, FD is 1310720
        via Connected, GigabitEthernet2

Device#

* EIGRP 100 is running on VRF 2.
------------------------------------------------------------------------------
```

View EIGRP Route After BGP Redistribution

Use the **show eigrp address-family ipv4 vrf**<*vrf-num*>**topology** command to view the BGP route is redistributed into the EIGRP protocol.

```
Device#show eigrp address-family ipv4 vrf 2 topology
EIGRP-IPv4 VR(test) Topology Table for AS(100)/ID(10.10.10.2)
           Topology(base) TID(0) VRF(2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.10.10.0/8, 1 successors, FD is 1310720
        via Connected, GigabitEthernet2
P 172.16.0.0/12, 1 successors, FD is 131072000
        via +Redistributed (131072000/0)

-Device#
```

```
* BGP route has been redistributed into EIGRP.
```

Examples: Configure Route Redistribution

The following is a sample configuration for configuring route redistribution between a global VRF and service VPN.

In this example, VRF 103 and VRF 104 are the service VPNs. The example shows that BGP routes are redistributed from the global VRF to VRF 103 and VRF 104.

```
router bgp 100
 address-family ipv4 vrf 103
  redistribute vrf global bgp 100 route-map test2
!
 address-family ipv4 vrf 104
  redistribute vrf global bgp 100 route-map test2
!
```

The following is a sample configuration for configuring the OSPF internal and external routes that are redistributed from the global VRF 65535 to the service VPN.

In this case, all the OSPF routes are redistributed into the service VPN by using both the **internal** and **external** keywords.

Enter the commands in the configuration mode as follows:

```
router ospf 1
 redistribute vrf global ospf 65535 match internal external 1 external 2 subnets
```

The following is a sample configuration for configuring route redistribution from a service VPN to the global VRF .

```
router bgp 50000
 address-family ipv4
  redistribute vrf 102 bgp 50000 route-map test1
```

The following is a sample configuration for configuring route redistribution of BGP, connected, OSPF, and static protocols from the global VRF to VRF 1 when configuring under the EIGRP routing process.

```
router eigrp 101
 address-family ipv4 vrf 1
  redistribute vrf global bgp 50000 metric 1000000 10 255 1 1500
  redistribute vrf global connected metric 1000000 10 255 1 1500
  redistribute vrf global ospf 65535 match internal external 1 external 2 metric 1000000
10 255 1 1500
  redistribute vrf global static metric 1000000 10 255 1 1500
```

Verify Route Redistribution

The following example shows the output for the **show ip bgp** command using the **internal** keyword. This example shows a route from VRF 102 is redistributed successfully to the global VRF after the route is replicated.

```
Device# show ip bgp 10.10.10.10 internal

BGP routing table entry for 10.10.10.10/8, version 515
Paths: (1 available, best #1, table default)
Not advertised to any peer
```

```
Refresh Epoch 1
700000 70707
10.10.14.17 from 0.0.0.0 (172.16.255.15)
Origin IGP, aigp-metric 77775522, metric 7777, localpref 100, weight 32768, valid, sourced,
 replicated, best
Community: 0:7227 65535:65535
Extended Community: SoO:721:75 RT:50000:102
rx pathid: 0, tx pathid: 0x0
net: 0x7FB320235DC0, path: 0x7FB320245DF8, pathext: 0x7FB3203A4660
flags: net: 0x0, path: 0x808040003, pathext: 0x81
attribute: 0x7FB38E5B6258, ref: 14
Updated on Jul 1 2021 01:16:36 UTC
vm5#
```

The following example shows the output for the **show ip route** command to view the routes replicated for the redistribution.

```
Device# show ip route 10.10.10.10

Routing entry for 10.10.10.10/8
Known via "bgp 50000", distance 60, metric 7777
Tag 700000, type external,
replicated from topology(102)
Redistributing via ospf 65535, bgp 50000
Advertised by ospf 65535
bgp 50000 (self originated)
Last update from 10.10.14.17 5d15h ago
Routing Descriptor Blocks:
* 10.10.14.17 (102), from 10.10.14.17, 5d15h ago
opaque_ptr 0x7FB3202563A8
Route metric is 7777, traffic share count is 1
AS Hops 2
Route tag 700000
MPLS label: none
```

The following example shows the output for the **show ip bgp vpnv4 vrf** command using the **internal** keyword. In this output, the route is redistributed from the global VRF to VRF 102.

```
Device# show ip bgp vpnv4 vrf 102 209.165.201.0 internal

BGP routing table entry for 1:102:10.10.10.10/8, version 679
BGP routing table entry for 1:209.165.201.0/27, version 679
Paths: (1 available, best #1, table 102)
Advertised to update-groups:
4
Refresh Epoch 1
7111 300000
10.1.15.13 (via default) from 0.0.0.0 (172.16.255.15)
Origin IGP, aigp-metric 5755, metric 900, localpref 300, weight 32768, valid, sourced,
replicated, best
Community: 555:666
Large Community: 1:2:3 5:6:7 412789:412780:755
Extended Community: SoO:533:53 RT:50000:102
rx pathid: 0, tx pathid: 0x0
net: 0x7FB38E5C5718, path: 0x7FB3202668D8, pathext: 0x7FB38E69E960
flags: net: 0x0, path: 0x808040007, pathext: 0x181
attribute: 0x7FB320256798, ref: 7
Updated on Jul 6 2021 16:43:04 UTC
```

The following example show the output for the **show ip route vrf** *vrf-id* [**protocol**] command. In this output, you can view the leaked routes for redistribution.

```
Device# show ip route vrf 102 209.165.201.0

Routing Table: 102
```

```
Routing entry for 209.165.201.0/27
Known via "bgp 50000", distance 20, metric 900
Tag 7111, type external,
replicated from topology(default)
Redistributing via bgp 50000
Advertised by bgp 50000 (self originated)
Last update from 10.1.15.13 00:04:57 ago
Routing Descriptor Blocks:
* 10.1.15.13 (default), from 10.1.15.13, 00:04:57 ago
opaque_ptr 0x7FB38E5B5E98
Route metric is 900, traffic share count is 1
AS Hops 2
Route tag 7111
MPLS label: none
```

Examples: Configure Route Leaking and Redistribution

This example shows how to leak and redistribute routes from VRF 1 and VRF 2 into Global VRF.

```
configure terminal
 global-address-family ipv4 unicast
  route-replicate from vrf 1 unicast bgp 100 route-map foo1
  route-replicate from vrf 2 unicast bgp 100 route-map foo1

router bgp 100
 address-family ipv4
  redistribute vrf 1 bgp 100 route-map foo1
  redistribute vrf 2 bgp 100 route-map foo1
```

This example shows how to leak and redistribute routes from global VRF into VRF 1 and VRF 2.

```
configure terminal
 vrf definition 1
  address-family ipv4
   route-replicate from vrf global unicast bgp 100 route-map foo1

 vrf definition 2
  address-family ipv4
   route-replicate from vrf global unicast bgp 100 route-map foo1


  router bgp 100
   address-family ipv4 vrf 1
    redistribute bgp 100 route-map foo1

   address-family ipv4 vrf 2
    redistribute bgp 100 route-map foo1
```

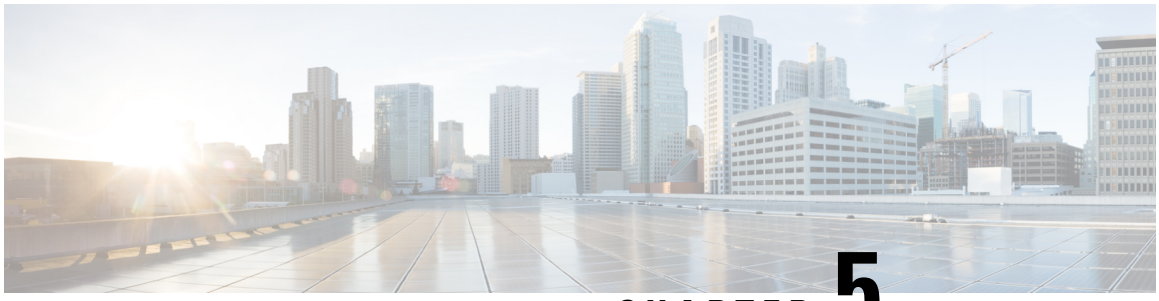
Feature Information for Route Leaking and Redistribution Between Global VRF and Service VPNs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information

Feature Name	Release Information	Description
Route Leaking Between Global VRF and Service VPNs	Cisco IOS XE Release 17.3.1a	This feature enables you to leak routes bidirectionally between the global VRF and service VPNs. Route leaking allows service sharing and is beneficial in migration use cases because it allows bypassing hubs and provides migrated branches direct access to non-migrated branches.
Redistribution of Replicated BGP Routes to OSPF, EIGRP Protocols	Cisco IOS XE Release 17.5.1a	This feature allows you to leak (or replicate) BGP routes between the global VRF and service VPNs, and redistribute the leaked BGP routes. The redistribution of the leaked routes to the EIGRP and OSPF protocols occurs after replicating the BGP routes into the corresponding VRF.
Redistribution of replicated routes into BGP	Cisco IOS XE Bengaluru Release 17.6.1	This feature allows you to leak (or replicate) routes between the global VRF and service VPNs, and redistribute the leaked routes into BGP. The redistribution of the leaked routes occurs after replicating the routes into the corresponding VRF.



CHAPTER 5

IPv4 Loop-Free Alternate Fast Reroute

When a link or a router fails, distributed routing algorithms compute new routes that take into account the failure. The time taken for computation is called routing transition. Until the transition is complete and all routers are converged on a common view of the network, the connectivity between the source and destination pairs is interrupted. You can use the IPv4 Loop-Free Alternate Fast Reroute feature to reduce the routing transition time to less than 50 milliseconds using a precomputed alternate next hop. When a router is notified of a link failure, the router immediately switches over to the repair path to reduce traffic loss.

IPv4 Loop-Free Alternate Fast Reroute supports the precomputation of repair paths. The repair path computation is done by the Intermediate System-to-Intermediate System (IS-IS) routing protocol, and the resulting repair paths are sent to the Routing Information Base (RIB). The repair path installation is done by Cisco Express Forwarding (formerly known as CEF) and Open Shortest Path First (OSPF).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for IPv4 Loop-Free Alternate Fast Reroute

- Loop-Free Alternate (LFA) Fast Reroute (FRR) can protect paths that are reachable through an interface only if the interface is a point-to-point interface.

- When a LAN interface is physically connected to a single neighbor, you should configure the LAN interface as a point-to-point interface so that it can be protected through LFA FRR.

Restrictions for IPv4 Loop-Free Alternate Fast Reroute

- A Multiprotocol Label Switching (MPLS) traffic engineering (TE) tunnel cannot be used as a protected interface. However, an MPLS TE tunnel can be a protecting (repair) interface as long as the TE tunnel is used as a primary path.
- Loadbalance support is available for FRR-protected prefixes, but the 50 ms cutover time is not guaranteed.
- A maximum of eight FRR-protected interfaces can simultaneously undergo a cutover.
- Only Layer 3 VPN is supported.
- IPv4 multicast is not supported.
- IPv6 is not supported.
- IS-IS will not calculate LFA for prefixes whose primary interface is a tunnel.
- LFA calculations are restricted to interfaces or links belonging to the same level or area. Hence, excluding all neighbors on the same LAN when computing the backup LFA can result in repairs being unavailable in a subset of topologies.
- Only physical and physical port-channel interfaces are protected. Subinterfaces, tunnels, and virtual interfaces are not protected.
- A TE label switched path (LSP) can be used as a backup path. However, the primary path has to be a physical interface, which can be used to achieve FRR in ring topologies.
- Border Gateway Protocol (BGP) Prefix-Independent Convergence (PIC) and IP FRR can be configured on the same interface as long as they are not used for the same prefix.

Information About IPv4 Loop-Free Alternate Fast Reroute

IS-IS and IP FRR

When a local link fails in a network, IS-IS recomputes new primary next-hop routes for all affected prefixes. These prefixes are updated in the RIB and the Forwarding Information Base (FIB). Until the primary prefixes are updated in the forwarding plane, traffic directed towards the affected prefixes are discarded. This process can take hundreds of milliseconds.

In IP FRR, IS-IS computes LFA next-hop routes for the forwarding plane to use in case of primary path failures. LFA is computed per prefix.

When there are multiple LFAs for a given primary path, IS-IS uses a tiebreaking rule to pick a single LFA for a primary path. In case of a primary path with multiple LFA paths, prefixes are distributed equally among LFA paths.

Repair Paths

Repair paths forward traffic during a routing transition. When a link or a router fails, due to the loss of a physical layer signal, initially, only the neighboring routers are aware of the failure. All other routers in the network are unaware of the nature and location of this failure until information about this failure is propagated through a routing protocol, which may take several hundred milliseconds. It is, therefore, necessary to arrange for packets affected by the network failure to be steered to their destinations.

A router adjacent to the failed link employs a set of repair paths for packets that would have used the failed link. These repair paths are used from the time the router detects the failure until the routing transition is complete. By the time the routing transition is complete, all routers in the network revise their forwarding data and the failed link is eliminated from the routing computation.

Repair paths are precomputed in anticipation of failures so that they can be activated the moment a failure is detected.

The IPv4 LFA FRR feature uses the following repair paths:

- Equal Cost Multipath (ECMP) uses a link as a member of an equal cost path-split set for a destination. The other members of the set can provide an alternative path when the link fails.
- LFA is a next-hop route that delivers a packet to its destination without looping back. Downstream paths are a subset of LFAs.

LFA Overview

LFA is a node other than the primary neighbor. Traffic is redirected to an LFA after a network failure. An LFA makes the forwarding decision without any knowledge of the failure.

An LFA must neither use a failed element nor use a protecting node to forward traffic. An LFA must not cause loops. By default, LFA is enabled on all supported interfaces as long as the interface can be used as a primary path.

Advantages of using per-prefix LFAs are as follows:

- The repair path forwards traffic during transition when the primary path link is down.
- All destinations having a per-prefix LFA are protected. This leaves only a subset (a node at the far side of the failure) unprotected.

LFA Calculation

The general algorithms to compute per-prefix LFAs can be found in RFC 5286. IS-IS implements RFC 5286 with a small change to reduce memory usage. Instead of performing a Shortest Path First (SPF) calculation for all neighbors before examining prefixes for protection, IS-IS examines prefixes after SPF calculation is performed for each neighbor. Because IS-IS examines prefixes after SPF calculation is performed, IS-IS retains the best repair path after SPF calculation is performed for each neighbor. IS-IS does not have to save SPF results for all neighbors.

Interaction Between RIB and Routing Protocols

A routing protocol computes repair paths for prefixes by implementing tiebreaking algorithms. The end result of the computation is a set of prefixes with primary paths, where some primary paths are associated with repair paths.

A tiebreaking algorithm considers LFAs that satisfy certain conditions or have certain attributes. When there is more than one LFA, configure the **fast-reroute per-prefix** command with the **tie-break** keyword. If a rule eliminates all candidate LFAs, then the rule is skipped.

A primary path can have multiple LFAs. A routing protocol is required to implement default tiebreaking rules and to allow you to modify these rules. The objective of the tiebreaking algorithm is to eliminate multiple candidate LFAs, select one LFA per primary path per prefix, and distribute the traffic over multiple candidate LFAs when the primary path fails.

Tiebreaking rules cannot eliminate all candidates.

The following attributes are used for tiebreaking:

- Downstream—Eliminates candidates whose metric to the protected destination is lower than the metric of the protecting node to the destination.
- Linecard-disjoint—Eliminates candidates sharing the same linecard with the protected path.
- Shared Risk Link Group (SRLG)—Eliminates candidates that belong to one of the protected path SRLGs.
- Load-sharing—Distributes remaining candidates among prefixes sharing the protected path.
- Lowest-repair-path-metric—Eliminates candidates whose metric to the protected prefix is higher.
- Node protecting—Eliminates candidates that are not node protected.
- Primary-path—Eliminates candidates that are not ECMPs.
- Secondary-path—Eliminates candidates that are ECMPs.

How to Configure IPv4 Loop-Free Alternate Fast Reroute

Configuring Fast Reroute Support

Note LFA computations are enabled for all routes, and FRR is enabled on all supported interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip router isis** *area-tag*
6. **isis tag** *tag-number*

7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip router isis** *area-tag*
11. **isis tag** *tag-number*
12. **exit**
13. **router isis** *area-tag*
14. **net** *net*
15. **fast-reroute per-prefix** {**level-1** | **level-2**} {**all** | **route-map** *route-map-name*}
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip router isis** *area-tag* **Example:** `Device(config-if)# ip router isis ipfrr`	Configures an IS-IS routing process for an IP on an interface and attaches an area designator to the routing process.
Step 6	**isis tag** *tag-number* **Example:** `Device(config-if)# isis tag 17`	Sets a tag on the IP address configured for an interface when the IP prefix is added to an IS-IS link-state packet (LSP).
Step 7	**exit** **Example:**	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 8	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/1`	Configures an interface and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.255.2 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 10	**ip router isis** *area-tag* **Example:** `Device(config-if)# ip router isis ipfrr`	Configures an IS-IS routing process for an IP on an interface and attaches an area designator to the routing process.
Step 11	**isis tag** *tag-number* **Example:** `Device(config-if)# isis tag 17`	Sets a tag on the IP address configured for an interface when the IP prefix is added to an IS-IS LSP.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 13	**router isis** *area-tag* **Example:** `Device(config)# router isis ipfrr`	Enables the IS-IS routing protocol, specifies an IS-IS process, and enters router configuration mode.
Step 14	**net** *net* **Example:** `Device(config-router)# net 49.0001.0101.2800.0001.00`	Configures an IS-IS network entity (NET) for a routing process.
Step 15	**fast-reroute per-prefix** {**level-1** | **level-2**} {**all** | **route-map** *route-map-name*} **Example:** `Device(config-router)# fast-reroute per-prefix level-2 all`	Enables per-prefix FRR. • Configure the **all** keyword to protect all prefixes.

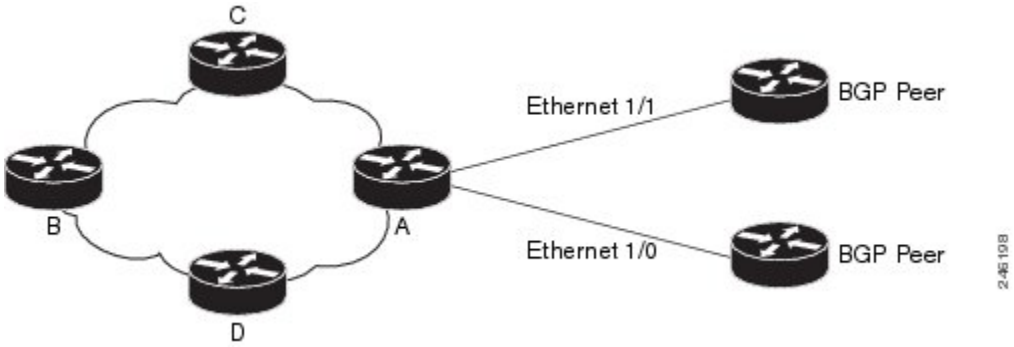
	Command or Action	Purpose
Step 16	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Configuration Examples for IPv4 Loop-Free Alternate Fast Reroute

Example: Configuring IPv4 Loop-Free Alternate Fast Reroute Support

The figure below shows IPv4 LFA FRR protecting BGP next hops by using interface tags.

Figure 8: Sample IPv4 LFA FRR Configuration



The following example shows how to configure IPv4 LFA FRR on Router A as shown in the above figure. Router A will advertise prefixes 10.0.0.0/24 and 192.168.255.0/24 along with the tag 17.

```
Device# configure terminal
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip router isis ipfrr
Device(config-if)# isis tag 17
Device(config-if)# exit
Device(config)# interface GigabitEthernet0/0/1
Device(config-if)# ip address 192.168.255.2 255.255.255.0
Device(config-if)# ip router isis ipfrr
Device(config-if)# isis tag 17
Device(config-if)# exit
Device(config)# router isis ipfrr
Device(config-router)# net 49.0001.0001.0001.0001.00
Device(config-router)# fast-reroute per-prefix level-2
```

The following example shows how to configure IPv4 LFA FRR on other routers as shown in the above figure. Other routers can use tag 17 to calculate repair paths for the two prefixes configured in Router A.

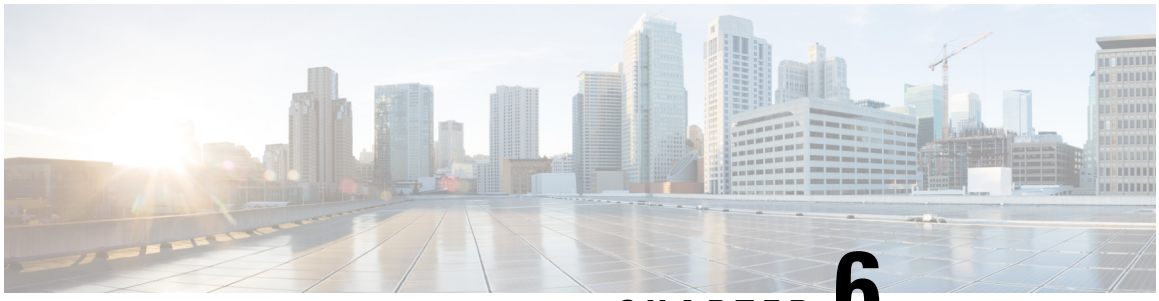
```
Device(config)# router isis
Device(config-router)# net 47.0004.004d.0001.0001.c11.1111.00
Device(config-router)# fast-reroute per-prefix level-2 route-map ipfrr-include
Device(config-router)# exit
Device(config)# route-map ipfrr-include
Device(config-router)# match tag 17
```

Feature Information for Configuring IPv4 Loop-Free Alternate Fast Reroute

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for Configuring IPv4 Loop-Free Alternate Fast Reroute



CHAPTER 6

IP Event Dampening

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for IP Event Dampening

Subinterface Restrictions

Only primary interfaces can be configured with this feature. The primary interface configuration is applied to all subinterfaces by default. IP Event Dampening does not track the flapping of individual subinterfaces on an interface.

Virtual Templates Not Supported

Copying a dampening configuration from virtual templates to virtual access interfaces is not supported because dampening has limited usefulness to existing applications that use virtual templates. Virtual access interfaces are released when an interface flaps, and new connections and virtual access interfaces are acquired when the interface comes up and is made available to the network. Since dampening states are attached to the interface, the dampening states would not survive an interface flap.

IPX Routing Protocols Not Supported

Internetwork Packet Exchange (IPX) protocols are not supported by the IP Event Dampening feature. However, IPX variants of these protocols will still receive up and down state event information when this feature is enabled. This should not create any problems or routing issues.

Information About IP Event Dampening

IP Event Dampening Overview

Interface state changes occur when interfaces are administratively brought up or down or if an interface changes state. When an interface changes state or flaps, routing protocols are notified of the status of the routes that are affected by the change in state. Every interface state change requires all affected devices in the network to recalculate best paths, install or remove routes from the routing tables, and then advertise valid routes to peer routers. An unstable interface that flaps excessively can cause other devices in the network to consume substantial amounts of system processing resources and cause routing protocols to lose synchronization with the state of the flapping interface.

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping. Dampening an interface removes the interface from the network until the interface stops flapping and becomes stable. Configuring the IP Event Dampening feature improves convergence times and stability throughout the network by isolating failures so that disturbances are not propagated. This, in turn, reduces the utilization of system processing resources by other devices in the network and improves overall network stability.

Interface State Change Events

This section describes the interface state change events of the IP Event Dampening features. This feature employs a configurable exponential decay mechanism that is used to suppress the effects of excessive interface flapping or state changes. When the IP Event Dampening feature is enabled, flapping interfaces are dampened from the perspective of the routing protocol by filtering excessive route updates. Flapping interfaces are identified, assigned penalties, suppressed if the necessary, and made available to the network when the interface stabilizes.

Suppress Threshold

The suppress threshold is the value of the accumulated penalty that triggers the router to dampen a flapping interface. The flapping interface is identified by the router and assigned a penalty for each up and down state change, but the interface is not automatically dampened. The router tracks the penalties that a flapping interface

accumulates. When the accumulated penalty reaches the default or preconfigured suppress threshold, the interface is placed in a dampened state.

Half-Life Period

The half-life period determines how fast the accumulated penalty can decay exponentially. When an interface is placed in a dampened state, the router monitors the interface for additional up and down state changes. If the interface continues to accumulate penalties and the interface remains in the suppress threshold range, the interface will remain dampened. If the interface stabilizes and stops flapping, the penalty is reduced by half after each half-life period expires. The accumulated penalty will be reduced until the penalty drops to the reuse threshold. The configurable range of the half-life period timer is from 1 to 30 seconds. The default half-life period timer is 5 seconds.

Reuse Threshold

When the accumulated penalty decreases until the penalty drops to the reuse threshold, the route is unsuppressed and made available to the other devices on the network. The range of the reuse value is from 1 to 20,000 penalties. The default value is 1000 penalties.

Maximum Suppress Time

The maximum suppress time represents the maximum amount of time an interface can remain dampened when a penalty is assigned to an interface. The maximum suppress time can be configured from 1 to 255 seconds. The default of the maximum penalty timer is 20 seconds or four times the default half-life period (5 seconds). The maximum value of the accumulated penalty is calculated, based on the maximum suppress time, reuse threshold, and half-life period.

Affected Components

When an interface is not configured with dampening, or when an interface is configured with dampening but is not suppressed, the routing protocol behavior as a result of interface state transitions is not changed by the IP Event Dampening feature. However, if an interface is suppressed, the routing protocols and routing tables are immune to any further state transitions of the interface until it is unsuppressed.

Route Types

The following interfaces are affected by the configuration of this feature:

- Connected routes:
  - The connected routes of dampened interfaces are not installed into the routing table.
  - When a dampened interface is unsuppressed, the connected routes will be installed into the routing table if the interface is up.
- Static routes:
  - Static routes assigned to a dampened interface are not installed into the routing table.
  - When a dampened interface is unsuppressed, the static route will be installed into the routing table if the interface is up.

Note Only the primary interface can be configured with this feature, and all subinterfaces are subject to the same dampening configuration as the primary interface. IP Event Dampening does not track the flapping of individual subinterfaces on an interface.

Supported Protocols

The IP Event Dampening feature supports Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Intermediate System-to-Intermediate System (IS-IS), Border Gateway Protocol (BGP), Connectionless Network Services (CLNS), and Hot Standby Routing Protocol (HSRP). The following list provides some general information about the operation of this feature with these protocols.

- RIP, OSPF, EIGRP, IS-IS, and BGP:
  - When an interface is dampened, the interface is considered to be down by the routing protocol. The routing protocol will not hold any adjacencies with this peer router over the dampened interface or generate advertisements of any routes related to this interface to other peer routers.
  - When the interface is unsuppressed and made available to the network, the interface will be considered by the routing protocols to be up. The routing protocols will be notified that the interface is in an up state and routing conditions will return to normal.
- HSRP:
  - When an interface is dampened, it is considered to be down by HSRP. HSRP will not generate HSRP messages out of the dampened interface or respond to any message received by the dampened interface. When the interface is unsuppressed and made available to the network, HSRP will be notified of the up state and will return to normal operations.
- CLNS:
  - When an interface is dampened, the interface is dampened to both IP and CLNS routing equally. The interface is dampened to both IP and CLNS because integrated routing protocols like IS-IS, IP, and CLNS routing are closely interconnected, so it is impossible to apply dampening separately.

Note The IP Event Dampening feature has no effect on any routing protocols if it is not enabled or an interface is not dampened.

Network Deployments

In real network deployments, some routers may not be configured with interface dampening, and all routers may not even support this feature. No major routing issues are expected, even if the router at the other end of a point-to-point interface or routers of the same multicast LAN do not have interface dampening turned on or do not have this feature implemented. On the router, where the interface is dampened, routes associated with the interface will not be used. No packets will be sent out of this interface, and no routing protocol activity will be initiated with routers on the other side of the interface. However, routers on the other side can still install some routes, in their routing tables, that are associated with this subnet because the routers recognize that their own interfaces are up and can start forwarding packets to the dampened interface. In such situations,

the router with the dampened interface will start forwarding these packets, depending on the routes in its routing table.

The IP Event Dampening feature does not introduce new information into the network. In fact, the effect of dampening is to subtract a subset of routing information from the network. Therefore, looping should not occur as a result of dampening.

Benefits of IP Event Dampening

Reduced Processing Load

The IP Event Dampening Feature employs a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols. Excessive interface up and down state changes that are received in a short period of time are not processed and do not consume system resources. Other routers in the network need not waste system resources because of a flapping route.

Faster Convergence

The IP Event Dampening feature improves convergence times and stability throughout the network by isolating failures so that disturbances are not propagated. Routers that are not experiencing link flap reach convergence sooner, because routing tables are not rebuilt each time the offending router leaves and enters the service

Improved Network Stability

The IP Event Dampening feature provides increased network stability. A router with a flapping interface removes the flapping interface from the network until the interface stabilizes, so other routers simply redirect traffic around the affected router until the interface becomes stable, which ensures that the router loses no data packets.

How to Configure IP Event Dampening

Enabling IP Event Dampening

The **dampening** command is entered in interface configuration mode to enable the IP Event Dampening feature. If this command is applied to an interface that already has dampening configured, all dampening states are reset and the accumulated penalty will be set to 0. If the interface has been dampened, the accumulated penalty will fall into the reuse threshold range, and the dampened interface will be made available to the network. The flap counts, however, are retained.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dampening** [*half-life-period reuse-threshold*] [*suppress-threshold max-suppress* [*restart-penalty*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface type number`	Enters interface configuration mode and configures the specified interface.
Step 4	**dampening** [*half-life-period reuse-threshold*] [*suppress-threshold max-suppress* [*restart-penalty*]] **Example:** `Router(config-if)# dampening`	Enables interface dampening. • Entering the **dampening** command without any arguments enables interface dampening with the default configuration parameters. • When manually configuring the timer for the *restart-penalty* argument, the values must be manually entered for all arguments.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Verifying IP Event Dampening

Use the **show dampening interface** or **show interface dampening** commands to verify the configuration of the IP Event Dampening feature.

The **clear counters** command may be used to clear the flap count and reset it to zero. All other parameters and status, including dampening states and accumulated penalties, are not affected by this command.

SUMMARY STEPS

1. **enable**
2. **show dampening interface**
3. **show interface dampening**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show dampening interface** **Example:** `Router# show dampening interface`	Displays dampened interfaces.
Step 3	**show interface dampening** **Example:** `Router# show interface dampening`	Displays dampened interfaces on the local router.

Configuration Examples for IP Event Dampening

Configuring IP Event Dampening Example

The following example configures interface dampening on Gigabit Ethernet interface 0/0/0 and sets the half life to 30 seconds, the reuse threshold to 1500, the suppress threshold to 10000, and the maximum suppress time to 120 seconds:

```
interface GigabitEthernet 0/0/0
 dampening 30 1500 10000 120
```

The following example configures interface dampening on ATM interface 2/0/0 and uses the default interface dampening values:

```
interface atm 2/0/0
 dampening
```

The following example configures the router to apply a penalty of 500 on Gigabit Ethernet interface 0/0/0 when the interface comes up for the first time after the router is reloaded:

```
interface GigabitEthernet 0/0/0
 dampening 5 500 1000 20 500
```

Verifying IP Event Dampening Example

The output of the **show dampening interface**command displays a summary of interface dampening.

```
Router# show dampening interface
3 interfaces are configured with dampening.
No interface is being suppressed.
```

```
Features that are using interface dampening:
  IP Routing
```

The output of the **show interface dampening** command displays the summary of the dampening parameters and the status of interfaces on the local router. The following is sample output from the **show interface dampening** command.

```
Router# show interface dampening
GigabitEthernet0/0/0
  Flaps Penalty    Supp ReuseTm   HalfL  ReuseV   SuppV  MaxSTm    MaxP Restart
      0       0   FALSE       0       5    1000    2000      20   16000       0
ATM2/0/0
  Flaps Penalty    Supp ReuseTm   HalfL  ReuseV   SuppV  MaxSTm    MaxP Restart
      0       0   FALSE       0       5    1000    2000      20   16000       0
POS2/0/0
  Flaps Penalty    Supp ReuseTm   HalfL  ReuseV   SuppV  MaxSTm    MaxP Restart
      0       0   FALSE       0       5    1000    2000      20   16000       0
```

Additional References

The following sections provide references related to the IP Event Dampening feature.

Related Documents

Related Topic	Document Title
IP Routing Protocol-Independent commands	*Cisco IOS IP Routing: Protocol-Independent Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for IP Event Dampening

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9: Feature Information for IP Event Dampening

Glossary

event dampening --The process in which a router dampens a flapping interface from the perspective of the routing tables and routing protocols of IP by filtering the excessive route adjust message because of the interface state change.

Flap --Rapid interface state changes from up to down and down to up within a short period of time.

half life --The rate of the exponential decay of the accumulated penalty is determined by this value.

maximum penalty --The maximum value beyond which the penalty assigned does not increase. It is derived from the maximum suppress time.

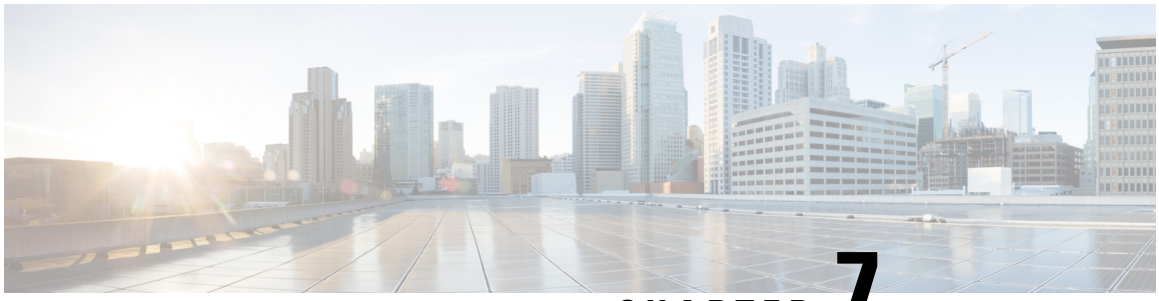
maximum suppress time --The maximum amount of time the interface can stay suppressed at the time a penalty is assigned.

penalty --A value assigned to an interface when it flaps. This value increases with each flap and decreases over time. The rate at which it decreases depends on the half life.

reuse threshold --The threshold value after which the interface will be unsuppressed and can be used again.

suppress threshold --Value of the accumulated penalty that triggers the router to dampen a flapping interface. When the accumulated penalty exceeds this value, the interface state is considered to be down from the perspective of the routing protocol.

suppressed --Suppressing an interface removes an interface from the network from the perspective of the routing protocol. An interface enters the suppressed state when it has flapped frequently enough for the penalty assigned to it to cross a threshold limit.



CHAPTER 7

PBR Recursive Next Hop

The PBR Recursive Next Hop feature enhances route maps to enable configuration of a recursive next-hop IP address that is used by policy-based routing (PBR). The recursive next-hop IP address is installed in the routing table and can be a subnet that is not directly connected. If the recursive next-hop IP address is not available, packets are routed using a default route.

Because Cisco Express Forwarding (CEF) or process switching provides the infrastructure, the benefit of this feature is the CEF loadsharing.

Restrictions for PBR Recursive Next Hop

If there are multiple equal-cost routes to the subnet that have been configured by the **set next-hop recursive** command, load balancing will occur only if all the adjacencies to the routes are resolved. If any of the adjacencies have not been resolved, load balancing will not occur and only one of the routes whose adjacency is resolved will be used. If none of the adjacencies are resolved, then the packets will be processed, resulting in the resolution of at least one of the adjacencies, leading to the programming of the adjacency in the hardware. Policy based routing relies on routing protocols or other means to resolve all adjacencies and as a result, load balancing occurs.

PBR Recursive Next Hope for IPv6 does not support load sharing.

Information About PBR Recursive Next-Hop

PBR Recursive Next Hop Overview

The PBR Recursive Next Hop feature enhances route maps to enable configuration of a recursive next-hop IP address that is used by policy-based routing (PBR). The recursive next-hop IP address is installed in the routing table and can be a subnet that is not directly connected. If the recursive next-hop IP address is not available, packets are routed using a default route.

PBR Recursive Next Hop for IPv6 also supports non-directly connected next hop. The recursive next hop specified can be a host address or a subnet address. The routing table is looked up to get the next hop based on the longest match of addresses. Only one such recursive next hop is supported per route map entry.

How to Configure PBR Recursive Next Hop

Setting the Recursive Next-Hop IP Address

The infrastructure provided by CEF or process switching performs the recursion to the next-hop IP address. The configuration sequence, which affects routing, is as follows:

1. Next-hop
2. Next-hop recursive
3. Interface
4. Default next-hop
5. Default interface

If both a next-hop address and a recursive next-hop IP address are present in the same route-map entry, the next hop is used. If the next hop is not available, the recursive next hop is used. If the recursive next hop is not available and no other IP address is present, the packet is routed using the default routing table; it is not dropped. If the packet is supposed to be dropped, use the **set ip next-hop**command with the **recursive** keyword, followed by a **set interface null0** configuration.

Perform this task to set the IP address for the recursive next-hop router.

Before you begin

If loadsharing is required, CEF loadsharing should be configured for per-packet or per-destination loadsharing. Loadbalancing should be done over all equal-cost routes to the subnet that has been configured by the **set ip next-hop recursive**command.

This functionality should be available in centralized and distributed systems.

Note Only one recursive next-hop IP address is supported per route-map entry.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source*[*source-wildcard*] [**log**]
4. **route-map** *map-tag*
5. Do one of the following:
   - **set ip next-hop** *ip-address*

- **set ipv6 next-hop** *ip-address*

6. Do one of the following:
   - **set ip next-hop** {*ip-address* [*...ip-address*] | **recursive** *ip-address*}
   - **set ipv6 next-hop** {*ipv6-address* [*...ipv6-address*] | **recursive** *ipv6-address*}
7. Do one of the following:
   - **match ip address** *access-list-number*
   - **match ipv6 address** {**prefix-list** *prefix-list-name* |*access-list-name*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source*[*source-wildcard*] [**log**] **Example:** `Router(config)# access-list 101 permit 10.60.0.0 0.0.255.255`	Configures an access list. The example configuration permits any source IP address that falls within the 10.60.0.0. 0.0.255.255 subnet.
Step 4	**route-map** *map-tag* **Example:** `Router(config)# route-map abccomp`	Enables policy routing and enters route-map configuration mode.

Step 5	Do one of the following:
	• **set ip next-hop** *ip-address*
	• **set ipv6 next-hop** *ip-address*
	Example:
	`Router(config-route-map)# set ip next-hop 10.10.1.1`
	Example:
	`Router(config-route-map)# set ipv6 next-hop 2001:DB8:2003:1::95`
		Sets a next-hop router IPv4 or IPv6 address.
		Note Set this IPv4/IPv6 address separately from the next-hop recursive router configuration.

Step 6	Do one of the following:	Sets a recursive next-hop IPv4/IPv6 address.

	Command or Action	Purpose

	• **set ip next-hop** {*ip-address* [*...ip-address*] | **recursive** *ip-address*}
	• **set ipv6 next-hop** {*ipv6-address* [*...ipv6-address*] | **recursive** *ipv6-address*}
	Example:
	`Router(config-route-map)# set ip next-hop recursive 10.20.3.3`
	Example:
	`Router(config-route-map)# set ipv6 next-hop recursive 2001:DB8:2003:2::95`
		Note This configuration does not ensure that packets get routed using the recursive IP address if an intermediate IP address is a shorter route to the destination.

Step 7	Do one of the following:
	• **match ip address** *access-list-number*
	• **match ipv6 address** {**prefix-list** *prefix-list-name* |*access-list-name*}
	Example:
	`Router(config-route-map)# match ip address 101`
	Example:
	`Router(config-route-map)# match ipv6 address kmd`
		Sets an access list to be matched.

Step 8	**end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Verifying the Recursive Next-Hop Configuration

To verify the recursive next-hop configuration, perform the following steps.

SUMMARY STEPS

1. **show running-config** | **begin abccomp**
2. **show route-map** *map-name*

DETAILED STEPS

Step 1 **show running-config** | **begin abccomp**

Use this command to verify the IPv4/IPv6 addresses for a next-hop and recursive next-hop IPv4/IPv6 address as listed in the following examples:

Example:

```
Router# show running-config | begin abccomp
route-map abccomp permit 10
 match ip address 101 ! Defines the match criteria for an access list.
 set ip next-hop recursive 10.3.3.3 ! If the match criteria are met, the recursive IP  address is
set.
 set ip next-hop 10.1.1.1 10.2.2.2 10.4.4.4
```

```
Router# show running-config | begin abccomp
route-map abccomp permit 10
 match ip address kmd! Defines the match criteria for an access list.
 set ipv6 next-hop recursive 2001:DB8:3000:1 ! If the match criteria are met, the recursive IPv6
address is set.
 set ipv6 next-hop 2001:DB8:3000:1 2001:DB8:4000:1 2001:DB8:5000:1
```

Step 2 **show route-map** *map-name*

Use this command to display the route maps, for example:

Example:

```
Router# show route-map abccomp
route-map abccomp, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip next-hop recursive 10.3.3.3
  ip next-hop 10.1.1.1 10.2.2.2 10.4.4.4
 Policy routing matches: 0 packets, 0 bytes
```

```
Router# show route-map abccomp
route-map abccomp, permit, sequence 10
 Match clauses:
  ipv6 address (access-lists): kmd
 Set clauses:
  ipv6 next-hop recursive 2001:DB8:3000:1
  ipv6 next-hop 2001:DB8:3000:1 2001:DB8:4000:1 2001:DB8:5000:1
 Policy routing matches: 0 packets, 0 bytes
```

Configuration Examples for PBR Recursive Next Hop

Example: Recursive Next-Hop IP Address

The following example shows the configuration of IP address 10.3.3.3 as the recursive next-hop router:

```
route-map abccomp
 set ip next-hop 10.1.1.1
 set ip next-hop 10.2.2.2
 set ip next-hop recursive 10.3.3.3
 set ip next-hop 10.4.4.4
```

The following example shows the configuration of IPv6 address 2001:DB8:2003:1::95 as the recursive next-hop router:

```
route-map abccomp
 set ipv6 next-hop 2001:DB8:2003:1::95
 set ipv6 next-hop 2001:DB8:2004:3::96
 set ipv6 next-hop recursive 2001:DB8:2005:2::95
 set ipv6 next-hop 2001:DB8:2006:1::95
```

Additional References for PBR Recursive Next Hop

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IP Routing: Protocol-Independent Command Reference
Performing basic system management	*Basic System Management Configuration Guide*
Changing the maximum number of paths	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN" module in the *BGP Configuration Guide*
BGP route map configuration tasks and configuration examples.	"Connecting to a Service Provider Using External BGP" module in the *BGP Configuration Guide*
BGP communities and route maps.	"BGP Cost Community" module in the *BGP Configuration Guide*
IPv6 Policy-Based Routing	"IPv6 Policy-Based Routing " module in the *IP Routing: Protocol-Independent Configuration Guide*

RFCs

RFC	Title
RFC 791	*Internet Protocol*
RFC 1219	*Variable-Length Subnet Masks*

Technical Assistance

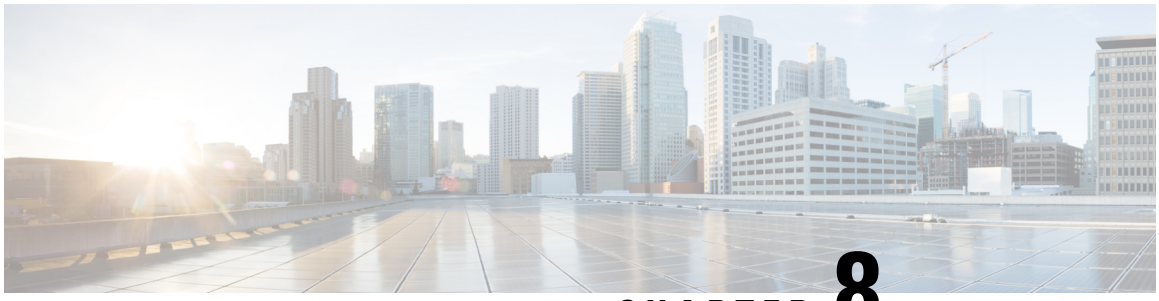
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for PBR Recursive Next Hop

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for PBR Recursive Next Hop



CHAPTER 8

PBR Support for Multiple Tracking Options

The PBR Support for Multiple Tracking Options feature extends the capabilities of object tracking using Cisco Discovery Protocol (CDP) to allow the policy-based routing (PBR) process to verify object availability by using additional methods. The verification method can be an Internet Control Message Protocol (ICMP) ping, a User Datagram Protocol (UDP) ping, or an HTTP GET request.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About PBR Support for Multiple Tracking Options

Object Tracking

Object tracking is an independent process that monitors objects such as the following:

- State of the line protocol of an interface
- Existence of an entry in the routing table
- Results of a Service Assurance Agent (SAA) operation, such as a ping

Clients such as Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), Gateway Load Balancing Protocol (GLBP), and (with this feature) PBR can register their interest in specific, tracked objects and then take action when the state of the objects changes.

PBR Support for Multiple Tracking Options Feature Design

The PBR Support for Multiple Tracking Options feature gives PBR access to all the objects that are available through the tracking process. The tracking process provides the ability to track individual objects--such as ICMP ping reachability, routing adjacency, an application running on a remote device, a route in the Routing Information Base (RIB)--or to track the state of an interface line protocol.

Object tracking functions in the following manner. PBR will inform the tracking process that a certain object should be tracked. The tracking process will in turn notify PBR when the state of that object changes.

How to Configure PBR Support for Multiple Tracking Options

Configuring PBR Support for Multiple Tracking Options

Perform this task to configure PBR support for multiple tracking options. In this task, a route map is created and configured to verify the reachability of the tracked object.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla monitor** *operation-number*
4. **type echo protocol ipIcmpEcho** {*destination-ip-address*| *destination-hostname*}[**source-ipaddr** {*ip-address*| *hostname*} | **source-interface** *interface-name*]
5. **exit**
6. **ip sla monitor schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**]
7. **track** *object-number* **rtr** *entry-number* [**reachability**| **state**]
8. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask* [**secondary**]
12. **ip policy route-map** *map-tag*
13. **exit**
14. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
15. **set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*]
16. **end**
17. **show track** *object-number*
18. **show route-map** [*map-name*| **all**| **dynamic**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla monitor** *operation-number* **Example:** `Device(config)# ip sla monitor 1`	Starts a Cisco IOS IP Service Level Agreement (SLA) operation configuration and enters IP SLA monitor configuration mode.
Step 4	**type echo protocol ipIcmpEcho** {*destination-ip-address*| *destination-hostname*}[**source-ipaddr** {*ip-address*| *hostname*} | **source-interface** *interface-name*] **Example:** `Device(config-sla-monitor)# type echo protocol ipIcmpEcho 10.1.1.1`	Configures an IP SLA Internet Control Message Protocol (ICMP) echo probe operation.
Step 5	**exit** **Example:** `Device(config-sla-monitor)# exit`	Exits IP SLA monitor configuration mode and returns the device to global configuration mode.
Step 6	**ip sla monitor schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] [**recurring**] **Example:** `Device(config)# ip sla monitor schedule 1 life forever start-time now`	Configures the scheduling parameters for a single Cisco IOS IP SLA operation. • In this example, the time parameters for the IP SLA operation are configured.
Step 7	**track** *object-number* **rtr** *entry-number* [**reachability**| **state**] **Example:** `Device(config)# track 123 rtr 1 reachability`	Tracks the reachability of a Response Time Reporter (RTR) object and enters tracking configuration mode.
Step 8	**delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*} **Example:**	(Optional) Specifies a period of time, in seconds, to delay communicating state changes of a tracked object.

	Command or Action	Purpose
	`Device(config-track)# delay up 60 down 30`	
Step 9	**exit** **Example:** `Device(config-track)# exit`	Exits tracking configuration mode and returns the device to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface serial 2/0`	Specifies an interface type and number and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 192.168.1.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface. • See the "Configuring IPv4 Addresses" chapter of the *Cisco IOS IP Addressing Services Configuration Guide* for information on configuring IPv4 addresses. • In this example, the IP address of the incoming interface is specified. This is the interface on which policy routing is to be enabled.
Step 12	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map alpha`	Enables policy routing and identifies a route map to be used for policy routing.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns the device to global configuration mode.
Step 14	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed.
Step 15	**set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*] **Example:** `Device(config-route-map)# set ip next-hop verify-availability 10.1.1.1 10 track 123`	Configures the route map to verify the reachability of the tracked object. • In this example, the policy is configured to forward packets received on serial interface 2/0 to 10.1.1.1 if that device is reachable.

	Command or Action	Purpose
Step 16	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns the device to privileged EXEC mode.
Step 17	**show track** *object-number* **Example:** `Device# show track 123`	(Optional) Displays tracking information. • Use this command to verify the configuration. See the display output in the "Examples" section of this task.
Step 18	**show route-map** [*map-name*| **all**| **dynamic**] **Example:** `Device# show route-map alpha`	(Optional) Displays route map information. • In this example, information about the route map named alpha is displayed. See the display output in the "Examples" section of this task.

Examples

The following output from the **show track** command shows that the tracked object 123 is reachable.

```
Device# show track 123
Track 123
  Response Time Reporter 1 reachability
  Reachability is Up
    2 changes, last change 00:00:33
  Delay up 60 secs, down 30 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 20
  Tracked by:
    ROUTE-MAP 0
```

The following output from the **show route-map** command shows information about the route map named alpha that was configured in the task.

```
Device# show route-map alpha
route-map alpha, permit, sequence 10
  Match clauses:
  Set clauses:
    ip next-hop verify-availability 10.1.1.1 10 track 123  [up]
  Policy routing matches: 0 packets, 0 bytes
```

Configuration Examples for PBR Support for Multiple Tracking Options

Example: Configuring PBR Support for Multiple Tracking Options

The following example shows how to configure PBR support for multiple tracking options.

The configured policy is that packets received on Ethernet interface 0, should be forwarded to 10.1.1.1 only if that device is reachable (responding to pings). If 10.1.1.1 is not up, then the packets should be forwarded to 10.2.2.2. If 10.2.2.2 is also not reachable, then the policy routing fails and the packets are routed according to the routing table.

Two RTRs are configured to ping the remote devices. The RTRs are then tracked. Policy routing will monitor the state of the tracked RTRs and make forwarding decisions based on their state.

```
! Define and start the RTRs.
ip sla monitor 1
 type echo protocol ipicmpecho 10.1.1.1
ip sla monitor schedule 1 start-time now life forever
!
ip sla monitor 2
 type echo protocol ipicmpecho 10.2.2.2
ip sla monitor schedule 2 start-time now life forever
!
! Track the RTRs.
track 123 rtr 1 reachability
track 124 rtr 2 reachability
!
! Enable policy routing on the incoming interface.
interface ethernet 0
 ip address 10.4.4.4 255.255.255.0
 ip policy route-map beta
!
! 10.1.1.1 is via this interface.
interface ethernet 1
 ip address 10.1.1.254 255.255.255.0
!
! 10.2.2.2 is via this interface.
interface ethernet 2
 ip address 10.2.2.254 255.255.255.0
!
! Define a route map to set the next-hop depending on the state of the tracked RTRs.
route-map beta
 set ip next-hop verify-availability 10.1.1.1 10 track 123
 set ip next-hop verify-availability 10.2.2.2 20 track 124
```

Additional References

The following sections provide references related to the PBR Support for Multiple Tracking Options feature.

Related Documents

Related Topic	Document Title
Object tracking within Cisco IOS software	Configuring Enhanced Object Tracking" chapter of the *Cisco IOS IP Application Services Configuration Guide*
Configuring IP addresses	"Configuring IPv4 Addresses" chapter of the *Cisco IOS IP Addressing Services Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IP Routing: Protocol-Independent Command Reference*. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases* , at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

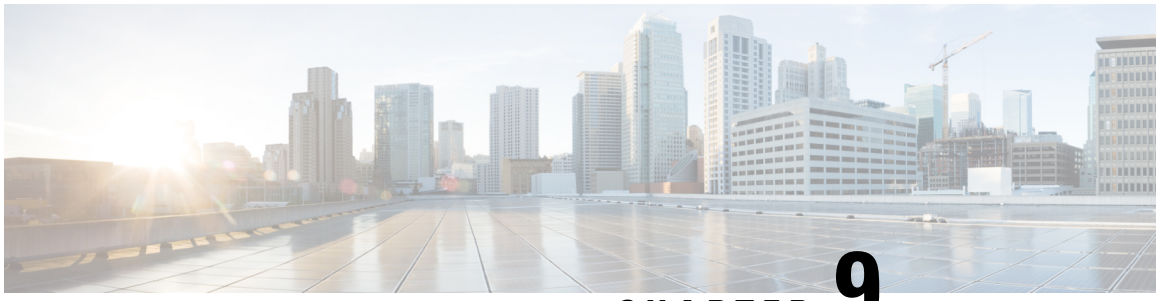
- **set ip next-hop verify-availability**

Feature Information for PBR Support for Multiple Tracking Options

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for PBR Support for Multiple Tracking Options



CHAPTER 9

PBR Match Track Object

The PBR Match Track Object feature enables a device to track the stub object during Policy Based Routing (PBR).

Restrictions for PBR Match Track Object

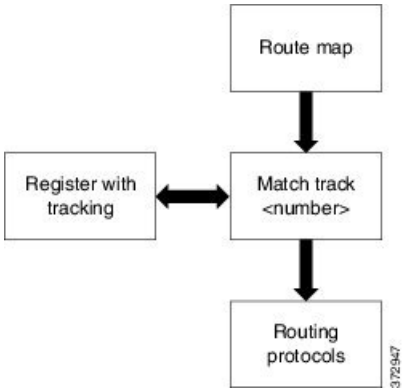
- You can use only one match track variable at a time in a route map sequence.
- You must remove the existing match track object configuration before configuring another match track object. The match track object is unregistered from the tracking component when you remove the match track object number configuration.
- Route-map for PBR, does not take ‘track-object’ into consideration when used under the ‘Match clause’. Match track-object is used for route distribution protocol (for example, BGP) only during the route distribution. Track object cannot be used in route-map, when that route-map is used in PBR.

Information About PBR Match Track Object

PBR Match Track Object Overview

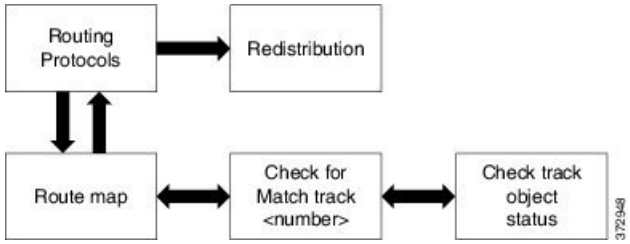
You refer to the stub object that you track as the match track object. The device checks for the existence of the match track object and issues an error message if there is none. Then registration with the tracking component is done to track this object. The device issues an error in case the registration fails.

Figure 9: Match track object registration



During redistribution, the routing protocols check the route map for matches with existing routes. This provides an exact route map that corresponds to the specific match criteria. When you apply this route map with the match track object, the device checks the status of the match track object and provides a specific route map.

Figure 10: Route map on redistribution using routing protocols



The device uses Border Gateway Protocol (BGP) for route-filtering and distribution. The device uses the existing notification mechanism to notify the routing protocols about the new match clause and also notifies the routing protocols about any change in the match track object status depending upon the Policy-Based Routing (PBR) query on redistribution.

How to Configure PBR Match Track Object

Configuring PBR Match Track Object

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag*
4. **match track** *track-object-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* **Example:** `Device(config)# route-map abc`	Enables policy routing and enters route-map configuration mode.
Step 4	**match track** *track-object-number* **Example:** `Device(config-route-map)# match track 2`	Tracks the stub object. Value ranges from 1 to 1000. **Note** This command is effective only when the track object specified is available on the device.
Step 5	**end** **Example:** `Device(config-route-map)# end`	Returns to privileged EXEC mode.

Verifying PBR Match Track Object

SUMMARY STEPS

1. **enable**
2. **show route-map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show route-map** *map-name* **Example:** `Device# show route-map abc`	Displays brief information about a specific route-map.

Configuration Examples for PBR Match Track Object

Example: PBR Match Track Object Configuration

```
Device> enable
Device# configure terminal
Device(config)# route-map abc
Device(config-route-map)# match track 2
Device(config-route-map)# end
```

Example: Verifying PBR Match Track Object

Sample output for the show route-map *map-name* command

To display information about a specific route-map, use the **show route-map** *map-name* command in privileged EXEC mode.

```
Device> enable
Device# show route-map abc
route-map abc, permit, sequence 10
  Match clauses:
    track-object 2
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
```

Additional References for PBR Match Track Object

Related Documents

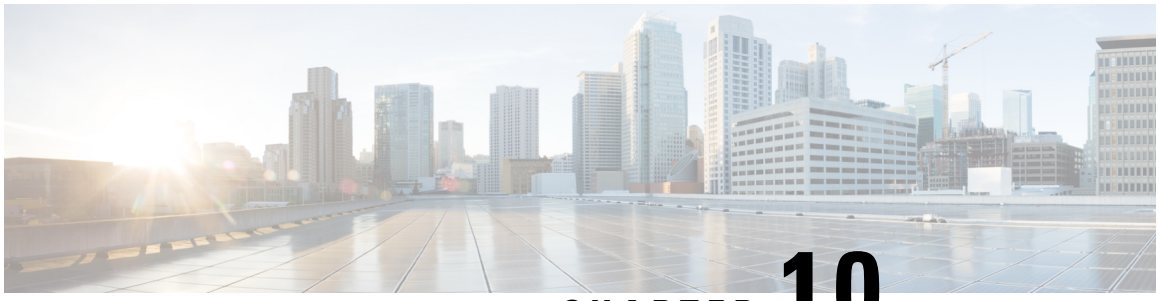
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for PBR Match Track Object

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 10

IPv6 Policy-Based Routing

Policy-based routing (PBR) in both IPv6 and IPv4 allows a user to manually configure how received packets should be routed. PBR allows the user to identify packets by using several attributes and to specify the next hop or the output interface to which the packet should be sent. PBR also provides a basic packet-marking capability.

Information About IPv6 Policy-Based Routing

Policy-Based Routing Overview

Policy-based routing (PBR) gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, which lessens reliance on routes derived from routing protocols. Therefore, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to set the IPv6 precedence. For a simple policy, you can use any one of these tasks; for a complex policy, you can use all of them. It also allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

PBR for IPv6 may be applied to both forwarded and originated IPv6 packets. For forwarded packets, PBR for IPv6 will be implemented as an IPv6 input interface feature, supported in the following forwarding paths:

- Process
- Cisco Express Forwarding (formerly known as CEF)
- Distributed Cisco Express Forwarding

Policies can be based on the IPv6 address, port numbers, protocols, or packet size.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, then, establish the match criteria.
- Set IPv6 precedence bits, giving the network the ability to enable differentiated classes of service.

- Route packets to specific traffic-engineered paths; you might need to route them to allow a specific quality of service (QoS) through the network.

PBR allows you to classify and mark packets at the edge of the network. PBR marks a packet by setting precedence value. The precedence value can be used directly by devices in the network core to apply the appropriate QoS to a packet, which keeps packet classification at your network edge.

How Policy-Based Routing Works

All packets received on an interface with policy-based routing (PBR) enabled are passed through enhanced packet filters called route maps. The route maps used by PBR dictate the policy, determining where to forward packets.

Route maps are composed of statements. The route map statements can be marked as permit or deny, and they are interpreted in the following ways:

- If a packet matches all match statements for a route map that is marked as permit, the device attempts to policy route the packet using the set statements. Otherwise, the packet is forwarded normally.
- If the packet matches any match statements for a route map that is marked as deny, the packet is not subject to PBR and is forwarded normally.
- If the statement is marked as permit and the packets do not match any route map statements, the packets are sent back through normal forwarding channels and destination-based routing is performed.

You must configure policy-based routing (PBR) on the interface that receives the packet, and not on the interface from which the packet is sent.

Packet Matching

Policy-based routing (PBR) for IPv6 will match packets using the **match ipv6 address** command in the associated PBR route map. Packet match criteria are those criteria supported by IPv6 access lists, as follows:

- Input interface
- Source IPv6 address (standard or extended access control list [ACL])
- Destination IPv6 address (standard or extended ACL)
- Protocol (extended ACL)
- Source port and destination port (extended ACL)
- DSCP (extended ACL)
- Flow-label (extended ACL)
- Fragment (extended ACL)

Packets may also be matched by length using the **match length** command in the PBR route map.

Match statements are evaluated first by the criteria specified in the **match ipv6 address** command and then by the criteria specified in the **match length** command. Therefore, if both an ACL and a length statement are used, a packet will first be subject to an ACL match. Only packets that pass the ACL match will be subject to the length match. Finally, only packets that pass both the ACL and the length statement will be policy routed.

Packet Forwarding Using Set Statements

Policy-based routing (PBR) for IPv6 packet forwarding is controlled by using a number of set statements in the PBR route map. These set statements are evaluated individually in the order shown, and PBR will attempt to forward the packet using each of the set statements in turn. PBR evaluates each set statement individually, without reference to any prior or subsequent set statement.

You may set multiple forwarding statements in the PBR for IPv6 route map. The following set statements may be specified:

- IPv6 next hop. The next hop to which the packet should be sent. The next hop must be present in the Routing Information Base (RIB), it must be directly connected, and it must be a global IPv6 address. If the next hop is invalid, the set statement is ignored.
- Output interface. A packet is forwarded out of a specified interface. An entry for the packet destination address must exist in the IPv6 RIB, and the specified output interface must be in the set path. If the interface is invalid, the statement is ignored.
- Default IPv6 next hop. The next hop to which the packet should be sent. It must be a global IPv6 address. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.
- Default output interface. The packet is forwarded out of a specified interface. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.

Note The order in which PBR evaluates the set statements is the order in which they are listed above. This order may differ from the order in which route-map set statements are listed by **show** commands.

When to Use Policy-Based Routing

Policy-based routing (PBR) can be used if you want certain packets to be routed some way other than the obvious shortest path. For example, PBR can be used to provide the following functionality:

- Equal access
- Protocol-sensitive routing
- Source-sensitive routing
- Routing based on interactive traffic versus batch traffic
- Routing based on dedicated links

Some applications or traffic can benefit from Quality of Service (QoS)-specific routing; for example, you could transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data such as e-mail over a lower-bandwidth, lower-cost link.

How to Enable IPv6 Policy-Based Routing

Enabling IPv6 PBR on an Interface

To enable Policy-Based Routing (PBR) for IPv6, you must create a route map that specifies the packet match criteria and desired policy-route action. Then you associate the route map on the required interface. All packets arriving on the specified interface that match the match clauses will be subject to PBR.

In PBR, the **set vrf** command decouples the virtual routing and forwarding (VRF) instance and interface association and allows the selection of a VRF based on access control list (ACL)-based classification using existing PBR or route-map configurations. It provides a single router with multiple routing tables and the ability to select routes based on ACL classification. The router classifies packets based on ACL, selects a routing table, looks up the destination address, and then routes the packet.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
4. Do one of the following:
   - **match length** *minimum-length maximum-length*
   - **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
5. Do one of the following:
   - **set ipv6 precedence** *precedence-value*
   - **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
   - **set interface** *type number* [*...type number*]
   - **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
   - **set default interface** *type number* [*...type number*]
   - **set vrf** *vrf-name*
6. **exit**
7. **interface** *type number*
8. **ipv6 policy route-map** *route-map-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. .

Step 4	Do one of the following:
	• **match length** *minimum-length maximum-length*
	• **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
	Example:
	`Device(config-route-map)# match length 3 200`
	Example:
	`Device(config-route-map)# match ipv6 address marketing`
		Specifies the match criteria.
		• You can specify any or all of the following:
		• Matches the Level 3 length of the packet.
		• Matches a specified IPv6 access list.
		• If you do not specify a **match** command, the route map applies to all packets.

Step 5	Do one of the following:
	• **set ipv6 precedence** *precedence-value*
	• **set ipv6 next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **set interface** *type number* [*...type number*]
	• **set ipv6 default next-hop** *global-ipv6-address* [*global-ipv6-address...*]
	• **set default interface** *type number* [*...type number*]
	• **set vrf** *vrf-name*
	Example:
	`Device(config-route-map)# set ipv6 precedence 1`
	Example:
	`Device(config-route-map)# set ipv6 next-hop 2001:DB8:2003:1::95`
	Example:
	`Device(config-route-map)# set interface GigabitEthernet 0/0/1`
	Example:
	`Device(config-route-map)# set ipv6 default next-hop 2001:DB8:2003:1::95`
		Specifies the action or actions to take on the packets that match the criteria.
		• You can specify any or all of the following:
		• Sets precedence value in the IPv6 header.
		• Sets next hop to which to route the packet (the next hop must be adjacent).
		• Sets output interface for the packet.
		• Sets next hop to which to route the packet, if there is no explicit route for this destination.
		• Sets output interface for the packet, if there is no explicit route for this destination.
		• Sets VRF instance selection within a route map for a policy-based routing VRF selection.

	Command or Action	Purpose

	Example:
	`Device(config-route-map)# set default interface GigabitEthernet 0/0/0`
	Example:
	`Device(config-route-map)# set vrf vrfname`	

Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 8	**ipv6 policy route-map** *route-map-name* **Example:** `Device(config-if)# ipv6 policy-route-map interactive`	Identifies a route map to use for IPv6 PBR on an interface.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling Local PBR for IPv6

Packets that are generated by the device are not normally policy routed. Perform this task to enable local IPv6 policy-based routing (PBR) for such packets, indicating which route map the device should use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 local policy route-map** *route-map-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 local policy route-map** *route-map-name* **Example:** `Device(config)# ipv6 local policy route-map pbr-src-90`	Configures IPv6 PBR for packets generated by the device.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying the Configuration and Operation of PBR for IPv6

SUMMARY STEPS

1. **enable**
2. **show ipv6 policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 policy** **Example:** `Device# show ipv6 policy`	Displays IPv6 policy routing packet activity.

Troubleshooting PBR for IPv6

Policy routing analyzes various parts of the packet and then routes the packet based on certain user-defined attributes in the packet.

SUMMARY STEPS

1. **enable**
2. **show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**] [**detailed**]
3. **debug ipv6 policy** [*access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**] [**detailed**] **Example:** `Device# show route-map`	Displays all route maps configured or only the one specified.
Step 3	**debug ipv6 policy** [*access-list-name*] **Example:** `Device# debug ipv6 policy`	Enables debugging of the IPv6 policy routing packet activity.

Configuration Examples for IPv6 Policy-Based Routing

Example: Enabling PBR on an Interface

In the following example, a route map named pbr-dest-1 is created and configured, specifying packet match criteria and desired policy-route action. PBR is then enabled on GigabitEthernet interface 0/0/1.

```
ipv6 access-list match-dest-1
  permit ipv6 any 2001:DB8:2001:1760::/32
route-map pbr-dest-1 permit 10
  match ipv6 address match-dest-1
  set interface GigabitEthernet 0/0/0
interface GigabitEthernet0/0/1
  ipv6 policy-route-map interactive
```

Example: Enabling Local PBR for IPv6

In the following example, packets with a destination IPv6 address that match the IPv6 address range allowed by access list pbr-src-90 are sent to the device at IPv6 address 2001:DB8:2003:1::95:

```
ipv6 access-list src-90
  permit ipv6 host 2001:DB8:2003::90 2001:DB8:2001:1000::/64
route-map pbr-src-90 permit 10
  match ipv6 address src-90
  set ipv6 next-hop 2001:DB8:2003:1::95
ipv6 local policy route-map pbr-src-90
```

Example: show ipv6 policy Command Output

The **show ipv6 policy** command displays PBR configuration, as shown in the following example:

```
Device# show ipv6 policy

Interface                 Routemap
GigabitEthernet0/0/0      src-1
```

Example: Verifying Route-Map Information

The following sample output from the **show route-map** command displays specific route-map information, such as a count of policy matches:

```
Device# show route-map

route-map bill, permit, sequence 10
  Match clauses:
  Set clauses:
  Policy routing matches:0 packets, 0 bytes
```

Additional References for IPv6 Policy-Based Routing

Related Documents

Related Topic	Document Title
IP Routing Protocol-Independent commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

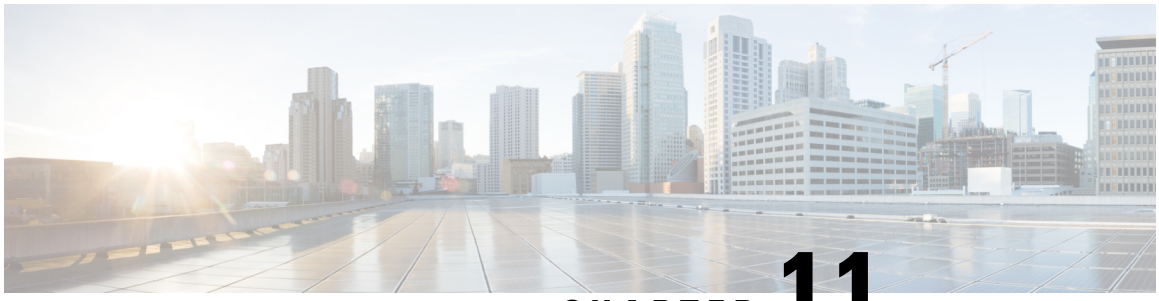
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Policy-Based Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 12: Feature Information for IPv6 Policy-Based Routing



CHAPTER 11

Multi-VRF Selection Using Policy-Based Routing

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) device to route packets to Virtual Private Networks (VPNs) based on packet length or match criteria defined in an IP access list.

You can enable VPN routing and forwarding (VRF) selection by policy routing packets through a route map, through the global routing table, or to a specified VRF.

You can enable policy-routing packets for VRF instances by using route map commands with **set** commands.

On supported hardware, you can configure both the Multi-VRF Selection Using Policy-Based Routing feature and the MPLS VPN VRF Selection Based on a Source IP Address feature on the same interface.

Prerequisites for Multi-VRF Selection Using Policy-Based Routing

- The device must support policy-based routing (PBR) in order for you to configure this feature. For platforms that do not support PBR, use the MPLS VPN VRF Selection Based on a Source IP Address feature.
- A Virtual Private Network (VPN) virtual routing and forwarding (VRF) instance must be defined before you configure this feature. An error message is displayed on the console if no VRF exists.

Restrictions for Multi-VRF Selection Using Policy-Based Routing

- All commands that aid in routing also support hardware switching, except for the **set ip next-hop verify availability** command because Cisco Discovery Protocol information is not available in the line cards.

- Protocol Independent Multicast (PIM) and multicast packets do not support policy-based routing (PBR) and cannot be configured for a source IP address that is a match criterion for this feature.
- The **set vrf** and **set ip global next-hop** commands can be configured with the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. But the **set vrf** and **set ip global next-hop** commands take precedence over the **set default interface**, **set interface** , **set ip default next-hop**, and **set ip next-hop** commands. No error message is displayed if you attempt to configure the **set vrf** command with any of these three **set** commands.
- The Multi-VRF Selection Using Policy-Based Routing feature cannot be configured with IP prefix lists.
- The **set global** and **set vrf** commands cannot be simultaneously applied to a route map.
- The Multi-VRF Selection Using Policy-Based Routing feature supports VRF-lite; that is, only IP routing protocols run on the device. Multiprotocol Label Switching (MPLS) and Virtual Private Networks (VPNs) cannot be configured. However, the **set vrf** command will work in MPLS VPN scenarios.
- If you delete one VRF using **no vrf definition** *vrf-name* command, then other VRFs in the VRF routing table are also removed unexpectedly; when **ip vrf receive** command is configured with receive entries above 400, and IPv4 and IPv6 routes above 2000.
- In a VRF receive scenario, the memory requirements are proportional to the number of VRF receives that are configured multiplied by the number of directly connected neighbours (Cisco Express Forwarding adjacencies). When the **ip vrf receive** command is configured, Cisco Express Forwarding adjacency prefixes are copied to the VRF. Network resources might be exhausted based on number of bytes per each adjacency prefix, number of adjacency prefixes, number of VRF receives configured, and the platform-specific route processor memory restrictions applicable to Cisco Express Forwarding entries.

Information About Multi-VRF Selection Using Policy-Based Routing

Policy Routing of VPN Traffic Based on Match Criteria

The Multi-VRF Selection Using Policy-Based Routing feature is an extension of the MPLS VPN VRF Selection Based on a Source IP Address feature. The Multi-VRF Selection Using Policy-Based Routing feature allows you to policy route Virtual Private Network (VPN) traffic based on match criteria. Match criteria are defined in an IP access list and/or are based on packet length. The following match criteria are supported in Cisco software:

- IP access lists—Define match criteria based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. All IP access list configuration options in Cisco software can be used to define match criteria.
- Packet lengths—Define match criteria based on the length of a packet, in bytes. The packet length filter is defined in a route map with the **match length** route-map configuration command.

Policy routing is defined in the route map. The route map is applied to the incoming interface with the **ip policy route-map** interface configuration command. An IP access list is applied to the route map with the **match ip address** route-map configuration command. Packet length match criteria are applied to the route map with the **match length** route-map configuration command. The **set** action is defined with the **set vrf**

route-map configuration command. The match criteria are evaluated, and the appropriate VRF is selected by the **set** command. This combination allows you to define match criteria for incoming VPN traffic and policy route VPN packets out to the appropriate virtual routing and forwarding (VRF) instance.

Policy-Based Routing set Commands

Policy-routing Packets for VRF Instances

To enable policy-routing packets for virtual routing and forwarding (VRF) instances, you can use route map commands with the following **set** commands. They are listed in the order in which the device uses them during the routing of packets.

- **set tos**—Sets the Type of Service (TOS) bits in the header of an IP packet.
- **set df**—Sets the Don't Fragment (DF) bit in the header of an IP packet.
- **set vrf**—Routes packets through the specified interface. The destination interface can belong only to a VRF instance. In case there is no route set for the destination in **set vrf** command routing will fall back to ingress interface VRF.
- **set global**—Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip vrf next-hop**—Indicates where to output IPv4 packets that pass a match criteria of a route map for policy routing when the IPv4 next hop must be under a specified VRF.
- **set ipv6 vrf next-hop**—Indicates where to output IPv6 packets that pass a match criteria of a route map for policy routing when the IPv6 next hop must be under a specified VRF.
- **set ip global next-hop**—Indicates where to forward IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table. The global keyword explicitly defines that IPv4 next-hops are under the global routing table.
- **set ipv6 global next-hop**—Indicates where to forward IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table. The global keyword explicitly defines that IPv6 next-hops are under the global routing table.
- **set interface**—When packets enter a VRF, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set ip default vrf**—Provides IPv4 inherit-VRF and inter-VRF routing. With inherit-VRF routing, IPv4 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv4 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ipv6 default vrf**—Provides IPv6 inherit-VRF and inter-VRF routing. With inherit-VRF routing, IPv6 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv6 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ip default global**—Provides IPv4 VRF to global routing.
- **set ipv6 default global**—Provides IPv6 VRF to global routing.
- **set default interface**—Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.

- **set ip default next-hop**—Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ipv6 default next-hop**—Indicates where to IPv6 output packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.

Change of Normal Routing and Forwarding Behavior

When you configure policy-based routing (PBR), you can use the following six **set** commands to change normal routing and forwarding behavior. Configuring any of these **set** commands, with the potential exception of the **set ip next-hop** command, overrides the routing behavior of packets entering the interface if the packets do not belong to a virtual routing and forwarding (VRF) instance. The packets are routed from the egress interface across the global routing table.

- **set default interface**—Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination.
- **set interface**—When packets enter a VRF interface, routes the packets out of the egress interface under the same VRF according to the set interface policy, provided that the Layer 2 rewrite information is available.

Note The interface must be a peer-to-peer (P2P) interface.

- **set ip default next-hop**—Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ipv6 default next-hop**—Indicates where to output IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software has no explicit route to a destination.
- **set ip next-hop**—Indicates where to output IPv4 packets that pass a match criterion of a route map for policy routing. If an IPv4 packet is received on a VRF interface and is transmitted from another interface within the same VPN, the VRF context of the incoming packet is inherited from the interface.
- **set ipv6 next-hop**—Indicates where to output IPv6 packets that pass a match criterion of a route map for policy routing. If an IPv6 packet is received on a VRF interface and is transmitted from another interface within the same Virtual Private Network (VPN), the VRF context of the incoming packet is inherited from the interface.

Support of Inherit-VRF Inter-VRF and VRF-to-Global Routing

The Multi-VRF Selection Using Policy-Based Routing (PBR) feature supports inherit-VRF and inter-VRF. With inherit-VRF routing, packets arriving at a virtual routing and forwarding (VRF) interface are routed by the same outgoing VRF interface. With inter-VRF routing, packets arriving at a VRF interface are routed through any other outgoing VRF interface.

VRF-to-global routing causes packets that enter any VRF interface to be routed through the global routing table. When a packet arrives on a VRF interface, the destination lookup normally is done only in the corresponding VRF table. If a packet arrives on a global interface, the destination lookup is done in the global routing table.

The Multi-VRF Selection Using Policy-Based Routing feature modifies the following **set** commands to support inherit-VRF, inter-VRF, and VRF-to-global routing. The commands are listed in the order in which the device uses them during the routing of packets.

- **set global**—Routes packets through the global routing table. This command is useful for routing ingress packets belonging to a specific VRF through the global routing table.
- **set ip global next-hop**—Indicates where to forward IPv4 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table.
- **set ipv6 global next-hop**—Indicates where to forward IPv6 packets that pass a match criterion of a route map for policy routing and for which the Cisco software uses the global routing table.
- **set ip vrf next-hop**—Causes the device to look up the IPv4 next hop in the VRF table. If an IPv4 packet arrives on an interface that belongs to a VRF and the packet needs to be routed through a different VRF, you can use the **set ip vrf next-hop** command.
- **set ipv6 vrf next-hop**—Causes the device to look up the IPv6 next hop in the VRF table. If an IPv6 packet arrives on an interface that belongs to a VRF and the packet needs to be routed through a different VRF, you can use the **set ipv6 vrf next-hop** command.
- **set ip default vrf**—Provides IPv4 inherit-VRF and inter-VRF routing. With IPv4 inherit-VRF routing, IPv4 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv4 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set ipv6 default vrf**—Provides IPv6 inherit-VRF and inter-VRF routing. With IPv6 inherit-VRF routing, IPv6 packets arriving at a VRF interface are routed by the same outgoing VRF interface. With inter-VRF routing, IPv6 packets arriving at a VRF interface are routed through any other outgoing VRF interface.
- **set interface**—When packets enter a VRF, routes the packets out of the egress interface under the same VRF, according to the set interface policy, provided that the Layer 2 rewrite information is available.
- **set default interface**—Indicates where to output packets that pass a match criterion of a route map for policy routing and have no explicit route to the destination. The interface can belong to any VRF.
- **set ip next-hop**—Routes IPv4 packets through the global routing table in an IPv4-to-IPv4 routing and forwarding environment.
- **set ipv6 next-hop**—Routes IPv6 packets through the global routing table in an IPv6-to-IPv6 routing and forwarding environment.
- **set vrf**—Selects the appropriate VRF after a successful match occurs in the route map. VRS-aware PSV allows only inter-VRF (or VRF-to-VRF) switching.

How to Configure Multi-VRF Selection Using Policy-Based Routing

Defining the Match Criteria for Multi-VRF Selection Using Policy-Based Routing

Define the match criteria for the Multi-VRF Selection using Policy-Based Routing (PBR) feature so that you can selectively route the packets instead of using their default routing and forwarding.

The match criteria for the Multi-VRF Selection using Policy-Based Routing are defined in an access list. Standard, named, and extended access lists are supported.

You can define the match criteria based on the packet length by configuring the **match length** route-map configuration command. This configuration option is defined entirely within a route map.

The following sections explain how to configure PBR route selection:

Configuring Multi-VRF Selection Using Policy-Based Routing with a Standard Access List

Before you begin

The tasks in the following sections assume that the virtual routing and forwarding (VRF) instance and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} [**source** *source-wildcard*] [**log**] **Example:** `Device(config)# access-list 40 permit source 10.1.1.0/24 0.0.0.255`	Creates an access list and defines the match criteria for the route map. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a standard access list numbered 40. This filter permits traffic from any host with an IP address in the 10.1.1.0/24 subnet.

Configuring Multi-VRF Selection Using Policy-Based Routing with a Named Extended Access List

To configure Multi-VRF Selection using Policy-Based Routing (PBR) with a named extended access list, complete the following steps.

Before you begin

The tasks in the following sections assume that the virtual routing and forwarding (VRF) instance and associated IP address are already defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*]
4. [*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** {**standard** | **extended**} [*access-list-name* | *access-list-number*] **Example:** `Device(config)# ip access-list extended NAMEDACL`	Specifies the IP access list type and enters the corresponding access list configuration mode. • You can specify a standard, extended, or named access list.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos***tos*] [**ttl** *operator-vaue*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit ip any any option any-options`	Defines the criteria for which the access list will permit or deny packets. • Match criteria can be defined based on IP addresses, IP address ranges, and other IP packet access list filtering options. Named, numbered, standard, and extended access lists are supported. You can use all IP access list configuration options to define match criteria. • The example creates a named access list that permits any configured IP option.

Configuring Multi-VRF Selection in a Route Map

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set** command configuration determines the VRF through which the outbound Virtual Private Network (VPN) packets will be policy routed.

Before you begin

You must define the virtual routing and forwarding (VRF) instance before you configure the route map; otherwise an error message appears on the console.

A receive entry must be added to the VRF selection table with the **ip vrf receive** command. If a match and set operation occurs in the route map but there is no receive entry in the local VRF table, the packet will be dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **named-ordering-route-map enable**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
5. Do one of the following :
   - **set ip vrf** *vrf-name* **next-hop** *global-ipv4-address* [*...global-ipv4-address*]
   - **set ipv6 vrf** *vrf-name* **next-hop** *global-ipv6-address* [*...global-ipv6-address*]
   - **set ip next-hop recursive vrf** *global-ipv4-address* [*...global-ipv4-address*]
   - **set ip global next-hop** *global-ipv4-address* [*...global-ipv4-address*]
   - **set ipv6 global next-hop** *global-ipv6-address* [*...global-ipv6-address*]
6. Do one of the following:
   - **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
   - **match length** *minimum-lengthmaximum-length*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**named-ordering-route-map enable**] **Example:**	Enables ordering of route-maps based on a string provided by the user.

	Command or Action	Purpose
	`Device(config)# named-ordering-route-map enable`	
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. .

Step 5	Do one of the following :
	• **set ip vrf** *vrf-name* **next-hop** *global-ipv4-address* [*...global-ipv4-address*]
	• **set ipv6 vrf** *vrf-name* **next-hop** *global-ipv6-address* [*...global-ipv6-address*]
	• **set ip next-hop recursive vrf** *global-ipv4-address* [*...global-ipv4-address*]
	• **set ip global next-hop** *global-ipv4-address* [*...global-ipv4-address*]
	• **set ipv6 global next-hop** *global-ipv6-address* [*...global-ipv6-address*]
	Example:
	`Device(config-route-map)# set ip vrf myvrf next-hop 10.0.0.0`
	Example:
	`Device(config-route-map)# set ipv6 vrf myvrf next-hop 2001.DB8:4:1::1/64`
	Example:
	`Device(config-route-map)# set ip next-hop recursive vrf 10.0.0.0`
	Example:
	`Device(config-route-map)# set ip global next-hop 10.0.0.0`
	Example:
	`Device(config-route-map)# set ipv6 global next-hop 2001.DB8:4:1::1/64`
		Indicates where to forward packets that pass a match criterion of a route map for policy routing when the IPv4 next hop must be under a specified VRF.
		Indicates where to forward packets that pass a match criterion of a route map for policy routing when the IPv6 next hop must be under a specified VRF.
		Indicates the IPv4 address to which destination or next hop is used for packets that pass the match criterion configured in the route map.
		Indicates the IPv4 address to forward packets that pass a match criterion of a route map for policy routing and for which the software uses the global routing table.
		Indicates the IPv6 address to forward packets that pass a match criterion of a route map for policy routing and for which the software uses the global routing table.

Step 6	Do one of the following: • **match ip address** {*acl-number* [*acl-name* | *acl-number*]} • **match length** *minimum-lengthmaximum-length* **Example:**	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. IP access lists are supported. • The example configures the route map to use standard access list 1 to define match criteria.

	Command or Action	Purpose

	`Device(config-route-map)# match ip address 1 or`
	Example:
	`Device(config-route-map)# match length 3 200`
		Specifies the Layer 3 packet length in the IP header as a match criterion in a class map.
		• The example configures the route map to match packets that are 3 to 200 bytes in length.

Step 7	**end** **Example:** `Device(config-route-map)# end`	Returns to privileged EXEC mode.

Configuring Multi-VRF Selection Using Policy-Based Routing and IP VRF Receive on the Interface

The route map is attached to the incoming interface with the **ip policy route-map** interface configuration command.

The source IP address must be added to the virtual routing and forwarding (VRF) selection table. VRF selection is a one-way (unidirectional) feature. It is applied to the incoming interface. If a **match** and **set** operation occurs in the route map but there is no receive entry in the local VRF table, the packet is dropped if the packet destination is local.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip vrf receive** *vrf-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface FastEthernet 0/1/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map map1`	Identifies a route map to use for policy routing on an interface. • The configuration example attaches the route map named map1 to the interface.
Step 5	**ip vrf receive** *vrf-name* **Example:** `Device(config-if)# ip vrf receive VRF-1`	Adds the IP addresses that are associated with an interface into the VRF table. • This command must be configured for each VRF that will be used for VRF selection.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying the Configuration of Multi-VRF Selection Using Policy-Based Routing

To verify the configuration of the Multi-VRF Selection Using Policy-Based Routing (PBR) feature, perform the following steps. You can enter the commands in any order.

SUMMARY STEPS

1. **show ip access-list** [*access-list-number* | *access-list-name*]
2. **show route-map** [*map-name*]
3. **show ip policy**

DETAILED STEPS

Step 1 **show ip access-list** [*access-list-number* | *access-list-name*]

Verifies the configuration of match criteria for Multi-VRF Selection Using Policy-Based Routing. The command output displays three subnet ranges defined as match criteria in three standard access lists:

Example:

```
Device# show ip access-list

Standard IP access list 40
 10 permit 10.1.0.0, wildcard bits 0.0.255.255
Standard IP access list 50
 10 permit 10.2.0.0, wildcard bits 0.0.255.255
```

```
Standard IP access list 60
 10 permit 10.3.0.0, wildcard bits 0.0.255.255
```

Step 2 **show route-map** [*map-name*]

Verifies **match** and **set** commands within the route map:

Example:

```
Device# show route-map
```

The output displays the match criteria and set action for each route-map sequence. The output also displays the number of packets and bytes that have been policy routed per each route-map sequence.

Example:

```
Device# show route-map map1

route-map map1, permit, sequence 10
Match clauses:
Set clauses:
 ip next-hop vrf myvrf 10.5.5.5  10.6.6.6  10.7.7.7
 ip next-hop global 10.8.8.8  10.9.9.9
Policy routing matches: 0 packets, 0 bytes
Device# show route-map map2
route-map map2, permit, sequence 10
Match clauses:
Set clauses:
 vrf myvrf
Policy routing matches: 0 packets, 0 bytes
Device# show route-map map3
route-map map3, permit, sequence 10
Match clauses:
Set clauses:
 global
Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip vrf next-hop** command:

Example:

```
Device(config)# route-map test

Device(config-route-map)# set ip vrf myvrf next-hop
Device(config-route-map)# set ip vrf myvrf next-hop 192.168.3.2
Device(config-route-map)# match ip address 255 101
Device(config-route-map)# end
Device# show route-map

route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip vrf myvrf next-hop 192.168.3.2
 Policy routing matches: 0 packets, 0 bytes
```

The following **show route-map** command displays output from the **set ip global** command:

Example:

```
Device(config)# route-map test
Device(config-route-map)# match ip address 255 101
Device(config-route-map)# set ip global next-hop 192.168.4.2
```

```
Device(config-route-map)# end
Device# show route-map

*May 25 13:45:55.551: %SYS-5-CONFIG_I: Configured from console by consoleout-map
route-map test, permit, sequence 10
 Match clauses:
  ip address (access-lists): 101
 Set clauses:
  ip global next-hop 192.168.4.2
 Policy routing matches: 0 packets, 0 bytes
```

Step 3 **show ip policy**

Verifies the Multi-VRF Selection Using Policy-Based Routing policy.

Example:

```
Device# show ip policy
```

The following **show ip policy** command output displays the interface and associated route map that is configured for policy routing:

Example:

```
Device# show ip policy

Interface                Route map
FastEthernet0/1/0        PBR-VRF-Selection
```

Configuration Examples for Multi-VRF Selection Using Policy-Based Routing

Example: Defining the Match Criteria for Multi-VRF Selection Using Policy-Based Routing

In the following example, three standard access lists are created to define match criteria for three different subnetworks. Any packets received on FastEthernet interface 0/1/0 will be policy routed through the PBR-VRF-Selection route map to the virtual routing and forwarding (VRF) that is matched in the same route-map sequence. If the source IP address of the packet is part of the 10.1.0.0/24 subnet, VRF1 will be used for routing and forwarding.

```
access-list 40 permit source 10.1.0.0 0.0.255.255
access-list 50 permit source 10.2.0.0 0.0.255.255
access-list 60 permit source 10.3.0.0 0.0.255.255
route-map PBR-VRF-Selection permit 10
 match ip address 40
 set vrf VRF1
 !
route-map PBR-VRF-Selection permit 20
 match ip address 50
 set vrf VRF2
 !
```

```
route-map PBR-VRF-Selection permit 30
 match ip address 60
 set vrf VRF3
 !
interface FastEthernet 0/1/0
 ip address 192.168.1.6 255.255.255.252
 ip policy route-map PBR-VRF-Selection
 ip vrf receive VRF1
 ip vrf receive VRF2
 ip vrf receive VRF3
```

Example: Configuring Multi-VRF Selection in a Route Map

The following example shows a **set ip vrf next-hop** command that applies policy-based routing to the virtual routing and forwarding (VRF) interface named myvrf and specifies that the IP address of the next hop is 10.0.0.2:

```
Device(config)# route-map map1 permit
Device(config)# set vrf myvrf
Device(config-route-map)# set ip vrf myvrf next-hop 10.0.0.2
Device(config-route-map)# match ip address 101
Device(config-route-map)# end
```

The following example shows a **set ip global** command that specifies that the device should use the next hop address 10.0.0.1 in the global routing table:

```
Device(config-route-map)# set ip global next-hop 10.0.0.1
```

Additional References

Related Documents

Related Topic	Document Title
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
IP access list commands	*Cisco IOS Security Command Reference*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multi-VRF Selection Using Policy-Based Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13: Feature Information for Multi-VRF Selection Using Policy-Based Routing

Glossary

CE device—customer edge device. A device that is part of a customer network and that interfaces to a provider edge (PE) device.

Inherit-VRF routing—Packets arriving at a VRF interface are routed by the same outgoing VRF interface.

Inter-VRF routing—Packets arriving at a VRF interface are routed via any other outgoing VRF interface.

IP—Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

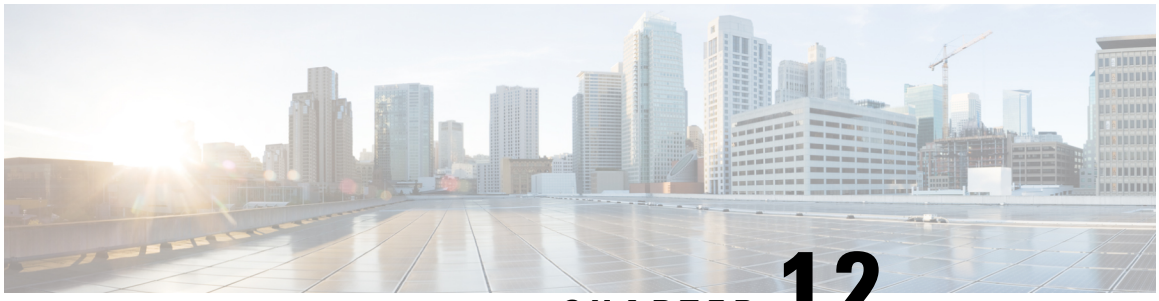
PBR—policy-based routing. PBR allows a user to manually configure how received packets should be routed.

PE device—provider edge device. A device that is part of a service provider's network and that is connected to a CE device. It exchanges routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.

VPN—Virtual Private Network. A collection of sites sharing a common routing table. A VPN provides a secure way for customers to share bandwidth over an ISP backbone network.

VRF—A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.

VRF-lite—A feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs.



CHAPTER 12

Multi-VRF Support

The Multi-VRF Support feature allows you to configure and maintain more than one instance of a routing and forwarding table within the same customer edge (CE) device.

Prerequisites for Multi-VRF Support

The network's core and provider edge (PE) devices must be configured for Virtual Private Network (VPN) operation.

Restrictions for Multi-VRF Support

- You can configure the Multi-VRF Support feature only on Layer 3 interfaces.
- The Multi-VRF Support feature is not supported by Interior Gateway Routing Protocol (IGRP) nor Intermediate System to Intermediate System (IS-IS).
- Label distribution for a given VPN routing and forwarding (VRF) instance on a given device can be handled by either Border Gateway Protocol (BGP) or Label Distribution Protocol (LDP), but not by both protocols at the same time.
- Multicast cannot operate on a Layer 3 interface that is configured with the Multi-VRF Support feature.
- MPLS IP Rewrite Manager (IPRM) does not support Equal Cost Multi-Path (ECMP) on the default VRF route.

Information About Multi-VRF Support

How the Multi-VRF Support Feature Works

The Multi-VRF Support feature enables a service provider to support two or more Virtual Private Networks (VPNs), where the IP addresses can overlap several VPNs. The Multi-VRF Support feature uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each virtual routing and forwarding (VRF) instance. Interfaces in a VRF can be either physical, such as FastEthernet ports, or logical, such as VLAN , but a Layer 3 interface cannot belong to more than one VRF at any one time. The Multi-VRF Support feature allows an operator to support two or more routing domains on a customer edge (CE) device, with each routing domain having its own set of interfaces and its own set of routing and forwarding tables. The Multi-VRF Support feature makes it possible to extend the label switched paths (LSPs) to the CE and into each routing domain that the CE supports.

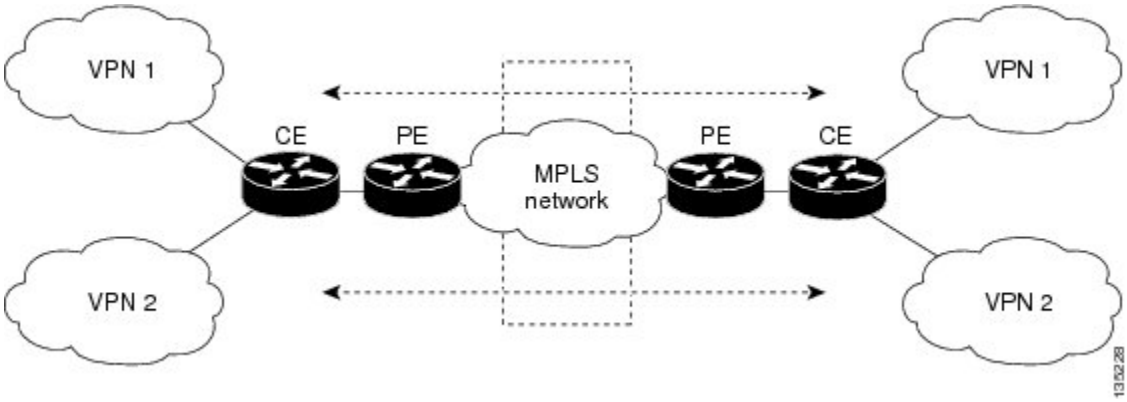
The Multi-VRF Support feature works as follows:

- Each CE device advertises its site's local routes to a provider edge (PE) device and learns the remote VPN routes from that provider edge (PE) device.
- PE devices exchange routing information with CE devices by using static routing or a routing protocol such as the Border Gateway Protocol (BGP), Routing Information Protocol version 1 (RIPv1), or RIPv2.
- PE devices exchange MPLS label information with CE devices through Label Distribution Protocol (LDP) or BGP.
- The PE device needs to maintain VPN routes only for those VPNs to which it is directly attached, eliminating the requirement that the PE maintain all of the service provider's VPN routes. Each PE device maintains a VRF for each of its directly connected sites. Two or more interfaces on a PE device can be associated with a single VRF if all the sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CE devices, the PE device exchanges VPN routing information with other PE devices through internal BGP (iBGP).

With the Multi-VRF Support feature, two or more customers can share one CE device, and only one physical link is used between the CE and the PE devices. The shared CE device maintains separate VRF tables for each customer and routes packets for each customer based on that customer's own routing table. The Multi-VRF Support feature extends limited PE device functionality to a CE device, giving it the ability, through the maintenance of separate VRF tables, to extend the privacy and security of a VPN to the branch office.

The figure below shows a configuration where each CE device acts as if it were two CE devices. Because the Multi-VRF Support feature is a Layer 3 feature, each interface associated with a VRF must be a Layer 3 interface.

Figure 11: Each CE Device Acting as Several Virtual CE Devices



How Packets Are Forwarded in a Network Using the Multi-VRF Support Feature

Following is the packet-forwarding process in an Multi-VRF customer edge (CE)-enabled network, as illustrated in the figure above:

- When the CE receives a packet from a Virtual Private Network (VPN), it looks up the routing table based on the input interface. When a route is found, the CE imposes the Multiprotocol Label Switching (MPLS) label that it received from the provider edge (PE) for that route and forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it swaps the incoming label with the corresponding label stack and sends the packet to the MPLS network.
- When an egress PE receives a packet from the network, it swaps the VPN label with the label that it had earlier received for the route from the CE, and it forwards the packet to the CE.
- When a CE receives a packet from an egress PE, it uses the incoming label on the packet to forward the packet to the correct VPN.

To configure Multi-VRF, you create a VRF table and then specify the Layer 3 interface associated with that VRF. Next, you configure the routing protocols within the VPN, and between the CE and the PE. The Border Gateway Protocol (BGP) is the preferred routing protocol for distributing VPN routing information across the provider's backbone.

The Multi-VRF network has three major components:

- VPN route target communities: These are lists of all other members of a VPN community. You must configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE devices: This propagates VRF reachability information to all members of a VPN community. You must configure BGP peering in all PE devices within a VPN community.
- VPN forwarding: This transports all traffic between VPN community members across a VPN service-provider network.

Considerations When Configuring the Multi-VRF Support Feature

- A device with the Multi-VRF Support feature is shared by several customers, and each customer has its own routing table.
- Because each customer uses a different virtual routing and forwarding (VRF) table, the same IP addresses can be reused. Overlapping IP addresses are allowed in different Virtual Private Networks (VPNs).
- The Multi-VRF Support feature lets several customers share the same physical link between the provider edge (PE) and the customer edge (CE) devices. Trunk ports with several VLANs separate packets among the customers. Each customer has its own VLAN.
- For the PE device, there is no difference between using the Multi-VRF Support feature or using several CE devices.
- The Multi-VRF Support feature does not affect the packet-switching rate.

How to Configure Multi-VRF Support

Configuring VRFs

To configure virtual routing and forwarding (VRF) instances, complete the following procedure. Be sure to configure VRFs on both the provider edge (PE) and customer edge (CE) devices.

If a VRF has not been configured, the device has the following default configuration:

- No VRFs have been defined.
- No import maps, export maps, or route maps have been defined.
- No VRF maximum routes exist.
- Only the global routing table exists on the interface.

Note Multi-VRF/MVPN GRE configured layer-3 interface cannot participate in more than one VRF at the same time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **import map** *route-map*
8. **exit**
9. **interface** *type slot*/*subslot*/*port*[**.***subinterface*]

10. **ip vrf forwarding** *vrf-name*
11. **end**
12. **show ip vrf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Device(config)# ip routing`	Enables IP routing.
Step 4	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf v1`	Names the VRF, and enters VRF configuration mode.
Step 5	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates a VRF table by specifying a route distinguisher. Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).
Step 6	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:1`	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command works only if BGP is running.
Step 7	**import map** *route-map* **Example:** `Device(config-vrf)# import map importmap1`	(Optional) Associates a route map with the VRF.
Step 8	**exit** **Example:**	Returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-vrf)# exit`	
Step 9	**interface** *type slot/subslot/port*[**.***subinterface*] **Example:** `Device(config)# interface`	Specifies the Layer 3 interface to be associated with the VRF and enters interface configuration mode. The interface can be a routed port or an .
Step 10	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding v1`	Associates the VRF with the Layer 3 interface.
Step 11	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 12	**show ip vrf** **Example:** `Device# show ip vrf`	Displays the settings of the VRFs.

Configuring BGP as the Routing Protocol

Most routing protocols can be used between the customer edge (CE) and the provider edge (PE) devices. However, external BGP (eBGP) is recommended, because:

- BGP does not require more than one algorithm to communicate with many CE devices.
- BGP is designed to pass routing information between systems run by different administrations.
- BGP makes it easy to pass route attributes to the CE device.

When BGP is used as the routing protocol, it can also be used to handle the Multiprotocol Label Switching (MPLS) label exchange between the PE and CE devices. By contrast, if Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), Routing Information Protocol (RIP), or static routing is used, the Label Distribution Protocol (LDP) must be used to signal labels.

To configure a BGP PE-to-CE routing session, perform the following steps on the CE and on the PE devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **network** *ip-address* **mask** *network-mask*
5. **redistribute ospf** *process-id* **match internal**
6. **network** *ip-address wildcard-mask* **area** *area-id*
7. **address-family ipv4 vrf** *vrf-name*

8. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
9. **neighbor** *address* **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process with the autonomous system number passed to other BGP devices, and enters router configuration mode.
Step 4	**network** *ip-address* **mask** *network-mask* **Example:** `Device(config-router)# network 10.0.0.0 mask 255.255.255.0`	Specifies a network and mask to announce using BGP.
Step 5	**redistribute ospf** *process-id* **match internal** **Example:** `Device(config-router)# redistribute ospf 2 match internal`	Sets the device to redistribute OSPF internal routes.
Step 6	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Device(config-router)# network 10.0.0.0 255.255.255.0 area 0`	Identifies the network address and mask on which OSPF is running, and the area ID of that network address.
Step 7	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf v12`	Identifies the name of the virtual routing and forwarding (VRF) instance that will be associated with the next two commands, and enters VRF address-family mode.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:**	Informs this device's BGP neighbor table of the neighbor's address (or peer group name) and the neighbor's autonomous system number.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 10.0.0.3 remote-as 100`	
Step 9	**neighbor** *address* **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.3 activate`	Activates the advertisement of the IPv4 address-family neighbors.

Configuring PE-to-CE MPLS Forwarding and Signaling with BGP

If the Border Gateway Protocol (BGP) is used for routing between the provider edge (PE) and the customer edge (CE) devices, configure BGP to signal the labels on the virtual routing and forwarding (VRF) interfaces of both the CE and the PE devices. You must enable signalling globally at the router-configuration level and for each interface:

- At the router-configuration level, to enable Multiprotocol Label Switching (MPLS) label signalling via BGP, use the **neighbor send-label** command).
- At the interface level, to enable MPLS forwarding on the interface used for the PE-to-CE external BGP (eBGP) session, use the **mpls bgp forwarding** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **neighbor** *address* **send-label**
6. **neighbor** *address* **activate**
7. **end**
8. **configure terminal**
9. **interface** *type slot/subslot/port*[**.***subinterface*]
10. **mpls bgp forwarding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Configures the BGP routing process with the autonomous system number passed to other BGP devices and enters router configuration mode.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf v12`	Identifies the name of the VRF instance that will be associated with the next two commands and enters address family configuration mode.
Step 5	**neighbor** *address* **send-label** **Example:** `Device(config-router-af)# neighbor 10.0.0.3 send-label`	Enables the device to use BGP to distribute MPLS labels along with the IPv4 routes to the peer devices. If a BGP session is running when you issue this command, the command does not take effect until the BGP session is restarted.
Step 6	**neighbor** *address* **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.3 activate`	Activates the advertisement of the IPv4 address-family neighbors.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Returns to privileged EXEC mode.
Step 8	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 9	**interface** *type slot*/*subslot*/*port*[**.***subinterface*] **Example:** `Device(config)# interface`	Enters interface configuration mode for the interface to be used for the BGP session. The interface can be a routed port or an .
Step 10	**mpls bgp forwarding** **Example:** `Device(config-if)# mpls bgp forwarding`	Enables MPLS forwarding on the interface.

Configuring a Routing Protocol Other than BGP

You can use the Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), or static routing. This configuration uses OSPF, but the process is the same for other protocols.

If you use OSPF as the routing protocol between the provider edge (PE) and the customer edge (CE) devices, issue the **capability vrf-lite** command in router configuration mode.

Note If RIP EIGRP, OSPF or static routing is used, the Label Distribution Protocol (LDP) must be used to signal labels.

The Multi-VRF Support feature is not supported by Interior Gateway Routing Protocol (IGRP) or Intermediate System-to-Intermediate System (IS-IS).

Multicast cannot be configured on the same Layer 3 interface as the Multi-VRF Support feature is configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **log-adjacency-changes**
5. **redistribute bgp** *autonomous-system-number* **subnets**
6. **network** *ip-address subnet-mask* **area** *area-id*
7. **end**
8. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# router ospf 100 vrf v1`	Enables OSPF routing, specifies a virtual routing and forwarding (VRF) table, and enters router configuration mode.
Step 4	**log-adjacency-changes** **Example:**	(Optional) Logs changes in the adjacency state. This is the default state.

	Command or Action	Purpose
	`Device(config-router)# log-adjacency-changes`	
Step 5	**redistribute bgp** *autonomous-system-number* **subnets** **Example:** `Device(config-router)# redistribute bgp 800 subnets`	Sets the device to redistribute information from the Border Gateway Protocol (BGP) network to the OSPF network.
Step 6	**network** *ip-address subnet-mask* **area** *area-id* **Example:** `Device(config-router)# network 10.0.0.0 255.255.255.0 area 0`	Indicates the network address and mask on which OSPF runs, and the area ID of that network address.
Step 7	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 8	**show ip ospf** **Example:** `Device# show ip ospf`	Displays information about the OSPF routing processes.

Configuring PE-to-CE MPLS Forwarding and Signaling with LDP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot /subslot/port*[**.***subinterface*]
4. **mpls ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type slot /subslot/port*[**.***subinterface*] **Example:** `Device(config)# interface`	Enters interface configuration mode for the interface associated with the VRF. The interface can be a routed port or an .
Step 4	**mpls ip** **Example:** `Device(config-if)# mpls ip`	Enables MPLS forwarding of IPv4 packets along normally routed paths for this interface.

Configuration Examples for Multi-VRF Support

The figure below is an example of a Multi-VRF topology.

Example: Configuring Multi-VRF Support on the PE Device

The following example shows how to configure a VRF:

```
configure terminal
ip vrf v1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 exit
ip vrf v2
 rd 100:2
 route-target export 100:2
 route-target import 100:2
 exit
```

The following example shows how to configure on PE device, PE-to-CE connections using BGP for both routing and label exchange:

The following example shows how to configure on PE device, PE-to-CE connections using OSPF for routing and LDP for label exchange:

Example: Configuring Multi-VRF Support on the CE Device

The following example shows how to configure VRFs:

```
configure terminal
 ip routing
 ip vrf v11
  rd 800:1
  route-target export 800:1
  route-target import 800:1
  exit
 ip vrf v12
  rd 800:2
  route-target export 800:2
```

```
  route-target import 800:2
  exit
```

The following example shows how to configure CE device VPN connections:

```
interface
 ip vrf forwarding v11
 ip address 10.0.0.8 255.255.255.0
 exit
interface
 ip vrf forwarding v12
 ip address 10.0.0.8 255.255.255.0
 exit
router ospf 1 vrf v11
 network 10.0.0.0 255.255.255.0 area 0
 network 10.0.0.0 255.255.255.0 area 0
 exit
router ospf 2 vrf v12
 network 10.0.0.0 255.255.255.0 area 0
 network 10.0.0.0 255.255.255.0 area 0
 exit
```

Note If BGP is used for routing between the PE and CE devices, the BGP-learned routes from the PE device can be redistributed into OSPF using the commands in the following example.

```
router ospf 1 vrf v11
 redistribute bgp 800 subnets
 exit
router ospf 2 vrf v12
 redistribute bgp 800 subnets
 exit
```

The following example shows how to configure on CE devices, PE-to-CE connections using BGP for both routing and label exchange:

The following example shows how to configure on CE devices, PE-to-CE connections using OSPF for both routing and LDP for label exchange:

Additional References

Related Documents

Related Topic	Document Title
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference
OSPF with Multi-VRF	“OSPF Support for Multi-VRF in CE Routers” module in the OSPF Configuration Guide .

Technical Assistance

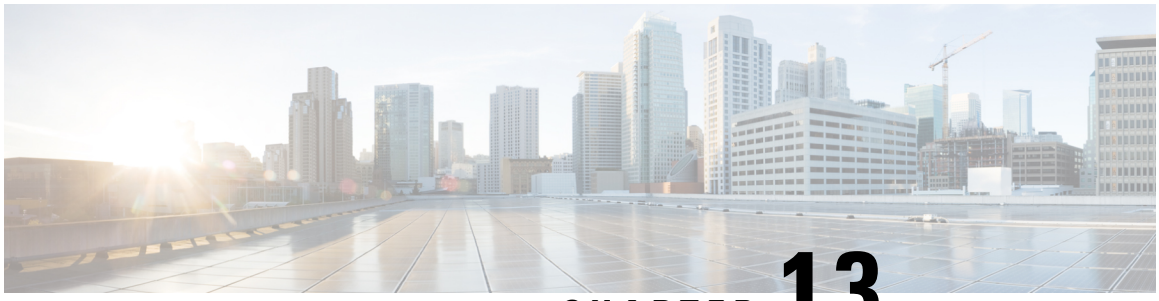
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multi-VRF Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 14: Feature Information for Multi-VRF Support



CHAPTER 13

Default Passive Interfaces

The Default Passive Interfaces feature simplifies the configuration of distribution devices by allowing all interfaces to be set as passive by default. In ISPs and large enterprise networks, many distribution devices have more than 200 interfaces. Obtaining routing information from these interfaces requires configuration of the routing protocol on all interfaces and manual configuration of the **passive-interface** command on interfaces where adjacencies were not desired.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Default Passive Interfaces

Default Passive Interfaces

In large enterprise networks, many distribution devices have more than 200 interfaces. Before the introduction of the Default Passive Interfaces feature, routing information could be obtained from these interfaces in these ways:

- Configure a routing protocol such as Open Shortest Path First (OSPF) on the backbone interfaces and redistribute connected interfaces.
- Configure a routing protocol on all interfaces and manually set most of them as passive.

Network operators might not always be able to summarize type 5 link-state advertisements (LSAs) at the device level where redistribution occurs, as in the first possibility. Thus, a large number of type 5 LSAs can be flooded over the domain.

In the second possibility, large type 1 LSAs might be flooded over the domain. The Area Border Router (ABR) creates type 3 LSAs, one for each type 1 LSA, and floods them to the backbone. You can, however, have unique summarization at the ABR level, which injects only one summary route into the backbone, thereby reducing the processing overhead.

Before the introduction of the Default Passive Interfaces feature, you could configure the routing protocol on all interfaces and manually set the **passive-interface** router configuration command on interfaces where adjacencies were not desired. But in some networks, this solution meant configuring 200 or more passive interfaces. The Default Passive Interfaces feature solved this problem by allowing all interfaces to be set as passive by default. You can set all interfaces as passive by default by using the **passive-interface default** command and then configure individual interfaces where adjacencies are desired using the **no passive-interface** command.

The Default Passive Interfaces feature simplifies the configuration of distribution devices and allows the network administrator to obtain routing information from interfaces in ISPs and large enterprise networks.

Preventing Routing Updates Through an Interface

To prevent other devices on a local network from learning about routes dynamically, you can keep routing update messages from being sent through a device interface. This feature applies to all IP-based routing protocols except the Border Gateway Protocol (BGP).

Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) behave somewhat differently. In OSPF, the interface address that you specify as passive appears as a stub network in the OSPF domain. OSPF routing information is neither sent nor received through the specified device interface. In IS-IS, the specified IP addresses are advertised without actually running IS-IS on those interfaces.

To prevent routing updates through a specified interface, use the **passive-interface** *type number* command in router configuration mode.

How to Configure Default Passive Interfaces

Configuring Default Passive Interfaces

Perform this task to set all interfaces on a device, in an Enhanced Interior Gateway Routing Protocol (EIGRP) environment, as passive by default, and then activate only those interfaces where adjacencies are desired.

Note When **passive-interface default** and **no-passive interface <int_name>** are configured, the **show run** command displays both interfaces. If you configure **passive-interface default** again, the **show run** command displays only the **passive-interface default**, and this causes the OSPF neighbors (if any) to flap. This behavior is specific to OSPF, and differs from other IGPs such as EIGRP and IS-IS.

In Cisco IOS XE 17.6.7, 17.9.5, 17.12.3, 17.14.x, and higher releases, this behavior has been modified for OSPF to be in line with EIGRP and IS-IS, i.e., when **passive-interface default** and **no-passive interface <int_name>** are configured, and you configure **passive-interface default** again, the **show run** command displays both interfaces, and OSPF neighbors do not flap.

This update is available on the following platforms:

- Cisco Catalyst 8500L Edge Platforms

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** {*autonomous-system-number* | *virtual-instance-number*}
4. **passive-interface** [**default**] [*type number*]
5. **no passive-interface** [**default**] [*type number*]
6. **network** *network-address* [*options*]
7. **end**
8. **show ip eigrp interfaces**
9. **show ip interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** {*autonomous-system-number* | *virtual-instance-number*} **Example:** `Device(config)# router eigrp 1`	Configures an EIGRP process and enters router configuration mode. • *autonomous-system-number*—Autonomous system number that identifies the services to the other EIGRP address-family devices. It is also used to tag routing information. The range is 1 to 65535.

	Command or Action	Purpose
		• *virtual-instance-number*—EIGRP virtual instance name. This name must be unique among all address-family router processes on a single device, but need not be unique among devices
Step 4	**passive-interface** [**default**] [*type number*] **Example:** `Device(config-router)# passive-interface default`	Sets all interfaces as passive by default.
Step 5	**no passive-interface** [**default**] [*type number*] **Example:** `Device(config-router)# no passive-interface gigabitethernet 0/0/0`	Activates only those interfaces that need adjacencies.
Step 6	**network** *network-address* [*options*] **Example:** `Device(config-router)# network 192.0.2.0`	Specifies the list of networks to be advertised by routing protocols.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 8	**show ip eigrp interfaces** **Example:** `Device# show ip eigrp interfaces`	Verifies whether interfaces on your network have been set to passive.
Step 9	**show ip interface** **Example:** `Device# show ip interface`	Verifies whether interfaces you enabled are active.

Configuration Examples for Default Passive Interfaces

Examples: Passive Interfaces Configuration for OSPF

In Open Shortest Path First (OSPF), hello packets are not sent on an interface that is specified as passive. Hence, the device is not able to discover any neighbors, and none of the OSPF neighbors are able to see the device on that network. In effect, this interface appears as a stub network to the OSPF domain. This configuration is useful if you want to import routes associated with a connected network into the OSPF domain without any OSPF activity on that interface.

The **passive-interface** router configuration command is typically used when the wildcard specification on the **network** router configuration command configures more interfaces than is desirable. The following configuration causes OSPF to run on all subnets of 172.18.0.0:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 172.18.1.1 255.255.255.0
Device(config-if)# exit
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# ip address 172.18.2.1 255.255.255.0
Device(config-if)# exit
Device(config)# interface GigabitEthernet 2/0/0
Device(config-if)# ip address 172.18.3.1 255.255.255.0
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# network 172.18.0.0 0.0.255.255 area 0
Device(config-router)# exit
```

If you do not want OSPF to run on 172.18.3.0, enter the following commands:

```
Device(config)# router ospf 1
Device(config-router)# network 172.18.0.0 0.0.255.255 area 0
Device(config-router)# no passive-interface GigabitEthernet 2/0/0
Device(config-router)# exit
```

Example: Default Passive Interfaces Configuration for OSPF

The following example configures the network interfaces, sets all interfaces that are running Open Shortest Path First (OSPF) as passive, and then enables serial interface 0/0/0:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 172.19.64.38 255.255.255.0 secondary
Device(config-if)# ip address 172.19.232.70 255.255.255.240
Device(config-if)# no ip directed-broadcast
Device(config-if)# exit
Device(config)# interface Serial 0/0/0
Device(config-if)# ip address 172.24.101.14 255.255.255.252
Device(config-if)# no ip directed-broadcast
Device(config-if)# no ip mroute-cache
Device(config-if)# exit
Device(config)# interface TokenRing 0/0/0
Device(config-if)# ip address 172.20.10.4 255.255.255.0
Device(config-if)# no ip directed-broadcast
Device(config-if)# no ip mroute-cache
Device(config-if)# ring-speed 16
Device(config-if)# exit
Device(config)# router ospf 1
Device(config-router)# passive-interface default
Device(config-router)# no passive-interface Serial 0/0/0
Device(config-router)# network 172.16.10.0 0.0.0.255 area 0
Device(config-router)# network 172.19.232.0 0.0.0.255 area 4
Device(config-router)# network 172.24.101.0 0.0.0.255 area 4
Device(config-router)# end
```

Additional References

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference

Technical Assistance

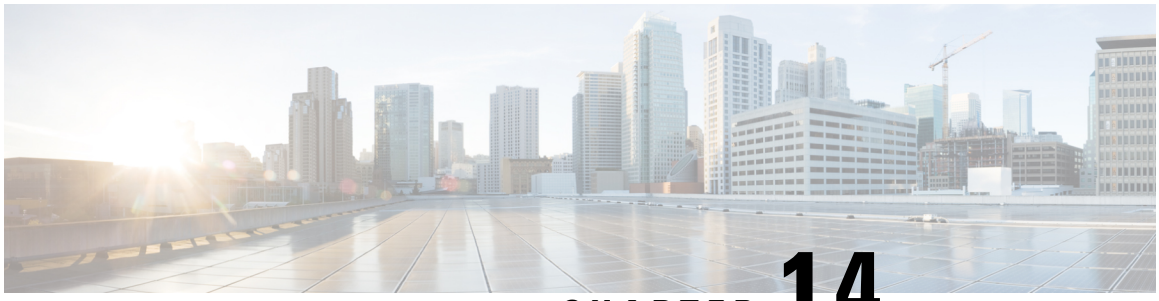
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Default Passive Interfaces

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 15: Feature Information for Default Passive Interfaces



CHAPTER 14

Policy-Based Routing

The Policy-Based Routing feature is a process whereby a device puts packets through a route map before routing the packets. The route map determines which packets are routed next to which device. Policy-based routing is a more flexible mechanism for routing packets than destination routing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Policy-Based Routing

For Policy-Based Routing, IPBase is a minimum licensing requirement.

Information About Policy-Based Routing

Policy-Based Routing

Policy-based routing (PBR) is a process whereby the device puts packets through a route map before routing them. The route map determines which packets are routed to which device next. You might enable policy-based

routing if you want certain packets to be routed some way other than the obvious shortest path. Possible applications for policy-based routing are to provide equal access, protocol-sensitive routing, source-sensitive routing, routing based on interactive versus batch traffic, and routing based on dedicated links. Policy-based routing is a more flexible mechanism for routing packets than destination routing.

To enable policy-based routing, you must identify which route map to use for policy-based routing and create the route map. The route map itself specifies the match criteria and the resulting action if all of the match clauses are met.

To enable policy-based routing on an interface, indicate which route map the device should use by using the **ip policy route-map** *map-tag* command in interface configuration mode. A packet arriving on the specified interface is subject to policy-based routing. This **ip policy route-map** command disables fast switching of all packets arriving on this interface.

To define the route map to be used for policy-based routing, use the **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq**] [*sequence-name* global configuration command.

To define the criteria by which packets are examined to learn if they will be policy-based routed, use either the **match length** *minimum-length maximum-length* command or the **match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*] command or both in route map configuration mode. No match clause in the route map indicates all packets.

To display the cache entries in the policy route cache, use the **show ip cache policy** command.

Note Mediatrace will show statistics of incorrect interfaces with policy-based routing (PBR) if the PBR does not interact with CEF or Resource Reservation Protocol (RSVP). Hence configure PBR to interact with CEF or RSVP directly so that mediatrace collects statistics only on tunnel interfaces and not physical interfaces.

Precedence Setting in the IP Header

The precedence setting in the IP header determines whether, during times of high traffic, the packets are treated with more or less precedence than other packets. By default, the Cisco software leaves this value untouched; the header remains with the precedence value that it had.

The precedence bits in the IP header can be set in the device when policy-based routing is enabled. When the packets containing those headers arrive at another device, the packets are ordered for transmission according to the precedence set, if the queueing feature is enabled. The device does not honor the precedence bits if queueing is not enabled; the packets are sent in FIFO order.

You can change the precedence setting, using either a number or name (the names came from RFC 791). You can enable other features that use the values in the **set ip precedence** route map configuration command to determine precedence. The table below lists the possible numbers and their corresponding name, from lowest to highest precedence.

Table 16: IP Precedence Values

Number	Name
0	routine
1	priority
2	immediate

Number	Name
3	flash
4	flash-override
5	critical
6	internet
7	network

The **set** commands can be used with each other. They are evaluated in the order shown in the previous table. A usable next hop implies an interface. Once the local device finds a next hop and a usable interface, it routes the packet.

Local Policy Routing

Packets that are generated by the device are not normally policy-routed. To enable local policy routing for such packets, indicate which route map the device should use by using the **ip local policy route-map** *map-tag* global configuration command. All packets originating on the device will then be subject to local policy routing.

Note Unlike UDP or other IP traffic, TCP traffic between a Cisco IOS or Cisco IOS-XE device and a remote host cannot be controlled using a local IP policy, if the Cisco device does not have an entry for the remote host IP in the Routing Information Base (RIB) (routing table) and Forwarding Information Base (FIB) (for Cisco Express Forwarding) . It is not necessary that the RIB or FIB entry should be the same path as the one being set by PBR. In the absence of this entry, TCP does not to detect a valid path to the destination and TCP traffic fails. However, UDP or ICMP traffic continues to be routed as per the local policy,

Use the **show ip local policy** command to display the route map used for local policy routing, if one exists.

How to Configure Policy-Based Routing

Configuring Policy-Based Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip policy route-map** *map-tag*
5. **exit**
6. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
7. Enter one or both of the following commands:

- **match length**
- **match ip address**

8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map equal-access`	Identifies a route map to use for policy routing on an interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 6	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. . • *map-tag*—A meaningful name for the route map. • **permit**—(Optional) If the match criteria are met for this route map, and the **permit** keyword is specified, the route is redistributed as controlled by the set actions. In the case of policy routing, the packet is policy routed. If the match criteria are not met, and the **permit** keyword is specified, the next route map with the same map tag is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set. • **deny**—(Optional) If the match criteria are met for the route map and the **deny** keyword is specified, the route is not redistributed. In the case of policy routing, the

	Command or Action	Purpose
		packet is not policy routed, and no further route maps sharing the same map tag name will be examined. If the packet is not policy routed, the normal forwarding algorithm is used. • *sequence-number*—(Optional) Number that indicates the position a new route map will have in the list of route maps already configured with the same name. If used with the **no** form of this command, the position of the route map **configure terminal** should be deleted.
Step 7	Enter one or both of the following commands: • **match length** • **match ip address** **Example:** `Device(config-route-map)# match ip address 1`	Define the criteria by which packets are examined to learn if they will be policy-based routed.
Step 8	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuration Examples for Policy-Based Routing

Additional References

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference

Technical Assistance

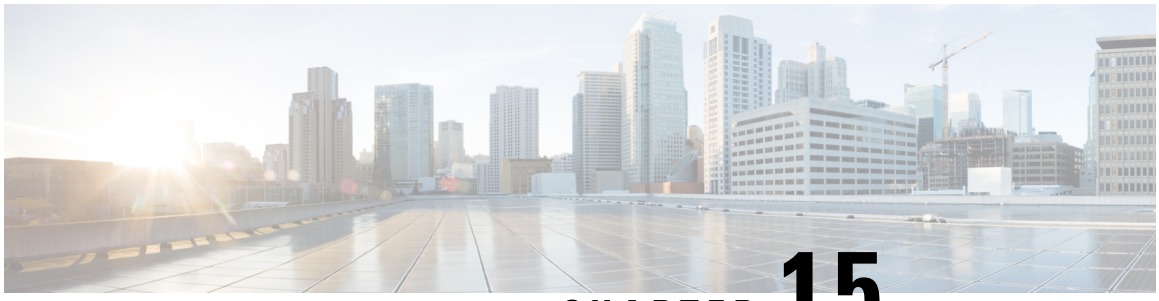
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Policy-Based Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 17: Feature Information for Policy-Based Routing



CHAPTER 15

Enhanced Policy-Based Routing and Site Manager

As network-based applications start being hosted on private or public cloud, network appliances forward network traffic based on configured policies. The enhanced Policy-based Routing (ePBR) routing enables application-based routing. Application-based routing provides a flexible, device-agnostic policy routing solution without impacting application performance.

Feature Information for ePBR - Application-Based Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 18: Feature Information for ePBR - Application-Based Routing

Information About Enhanced Policy-Based Routing and Site Manager

Restrictions for Enhanced Policy-Based Routing and Site Manager

- IPv6 is supported by enhanced policy-based routing but not supported by site manager
- Support is added only for ICMP probe, TCP probe is not supported

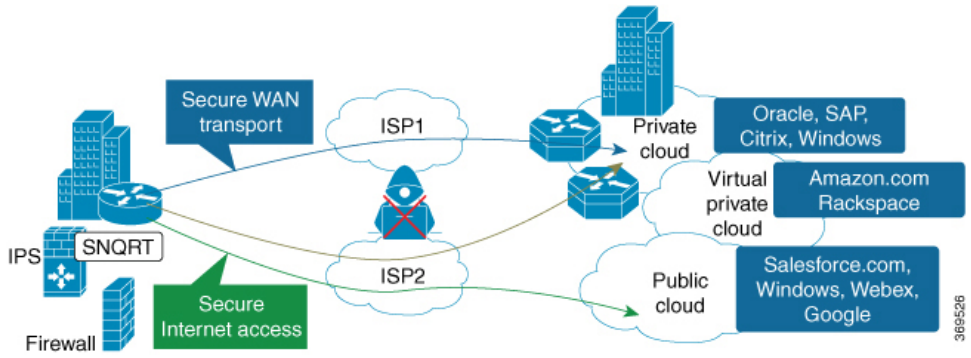
About Enhanced Policy-Based Routing and Site Manager

With central Internet access, all traffic traverses the Dynamic Multipoint VPN (DMVPN) tunnel and is routed to headquarters. This feature allows trusted SaaS traffic to be forwarded out over the optimized path (directly local break out) while other traffic still back-haul to headquarter over VPN.

Network-based Application Recognition version 2 (NBAR2) and Policy-Based Routing (PBR) solution first configures QoS to mark the SaaS application traffic to Differentiated Services Code Point (DSCP) 2, then configures PBR to redirect DSCP 2 traffic to Internet branch router DIA interface. However, this solution does not support flow stickness.

In the Enhanced Policy-Based Routing and Site Manager feature, using Site Manager Direct Cloud Access (DCA) and Direct Internet Access (DIA) you can selectively route cloud services applications such as Google, Salesforce, and Microsoft Office 365 through an Internet path that is specified in the path preference. Non-SaaS traffic can still be back-hauled to data center for further inspection.

Figure 12: Direct Cloud Access (DCA) / Direct Internet Access (DIA)



Site Manager

Site Manager and Border Router

- Site Manager—Site manager is a logical entity that implements specific policies on all border devices in a site. The site manager is also responsible for all policy-based routing and the path performance reported by border devices.

  This site manager has network connections to border routers and may connect to the centralized controller, if configured. You can define policies for the site manager or define policies in a centralized controller and publish to each site. Site-manager use default route as its nexthop address.

- Border Router—A border router is an enterprise WAN edge or internet edge device that connects to the site manager and gets routing information and reports path status. The border router forwards packets according to policy decision. Multiple border routers can be configured on one site and can be connected to the site controller.

The site manager is responsible for all policy-based routing and the path performance reported by a branch router.

Note NBAR classifcation occurs at branch router LAN ingress.

To achieve location proximity and to achieve better application performance, the SaaS server must be close to the branch router. Site Manager DCA uses cisco Umbrella branch to change DNS request from enterprise DNS resolver to a public DNS resolver, such as OpenDNS resolver or Google DNS resolver, which helps in placing the SaaS server closer to the branch router. OpenDNS account and registration is not manadatory. DNS request must be unencrypted traffic from the endpoint to the DNS server.

Prerequisites for Configuring Site Manager

- Cisco Umbrella branch must be enabled. Site Manager DCA uses a default route to determine the next-hop address, Cisco Umbrella is automatically enabled. For Site Manager DIA Cisco Umbrella branch must be enabled to intercept DNS to public DNS resolver.

Restrictions for Configuring Site Manager

- Site Manger does not support IPv6 addresses
- Site manager and Enhanced PBR may not work properly if NBAR does not classify packet properly.
- NBAR may not classify application properly in one of the following scenarios:
  - Proxy server is configured, or the DNS traffic does not pass through the router.
  - DNS request has encrypted traffic from the endpoint to the DNS server.

Feature Comparison

Feature/PBR	Application-Based Routing	Site Manager	Enhanced PBR
Flow Stickiness	Not Supported	Supported	Supported
Fallback Routing	EEM script to control the fallback routing	Path preference	
Symmetric	Asymmetric routing for dual branch scenario	Symmetric routing for dual branch scenario	

Benefits of ePBR – Application-Based Routing

- Directed Internet Access (DIA) – DIA routes Internet-bound traffic or public cloud traffic from the branch directly to the Internet. The ePBR-Application-based Routing feature allows you to local breakout guest Internet traffic and apply local security policies like Zone-based Firewall to the guest traffic.
- Directed Cloud Access (DCA) - To achieve improved Software as a Service (SaaS) application experience, you can define SaaS and its policy at the site manager. You can specify the DCA interfaces so that DCA path performance can be monitored and the best policy path can be selected. To achieve local proximity, the destination of the DNS request is modified to a public DNS resolver. The DNS request is then

forwarded through a DCA interface to an SaaS server close to the branch site, therefore achieving local breakout.

  - DNS request from end host is usually to an enterprise internal DNS server, in order to achieve location proximity, we modify the destination of the DNS request to a well-known public DNS resolver (like OpenDNS resolver, Google DNS resolver) and forward this DNS request through DCA interface, the DNS resolver gives a SaaS server close to the branch site, with this we usually can get a better SaaS application experience. You can also define local policy to merge with the global policy defined by the network hub, if IWAN is configured, or take precedence over the policy defined by hub, if IWAN is not configured.

- Flow-Stickness—Flow-stickness can provide first packet stickiness when NABR is applied. When the border router has multiple paths and a switch to a different path is triggered due to an event like performance downgrade, flow-stickness can keep the original path of traffic request stable connection.
- Outlook365 traffic category - Outlook365 endpoints are classified into optimize/allow/default by Microsoft based priority order and published externally. Network and security devices can apply its forwarding and security decision based on the traffic category information. Starting from IOS XE Amsterdam 17.3.1, SD-AVC gets traffic category information from Outlook365 and pushes to routing devices, ePBR can match this new traffic category attribute and policy routing to a certain path, for eg, bypass Firewall or local break out. This helps to improve Outlook365 application experience to avoid unnecessary latency due to security devices or backhaul path.
- Internet Edge Load Balancing - On the internet edge with multiple ISP links, you can use advanced load balance algorithms, such as static weight, and dynamic link bandwidth per packet to forward specific traffic to one ISP or load balance among the existing ISP links. This helps to fully utilize all the available links instead of only the active or backup links. On the internet edge with multiple ISP links, you can define apolicy to forward specific traffic to one ISP or load balance among the existing ISP links.

Configure Enhanced PBR to Allow and Optimize Office365 Traffic

Enable SD-AVC on the devices to get the traffic category information from Office365 cloud, you can then enable Enhanced Policy-based Routing (ePBR) to steer Office365 traffic to the expected path.

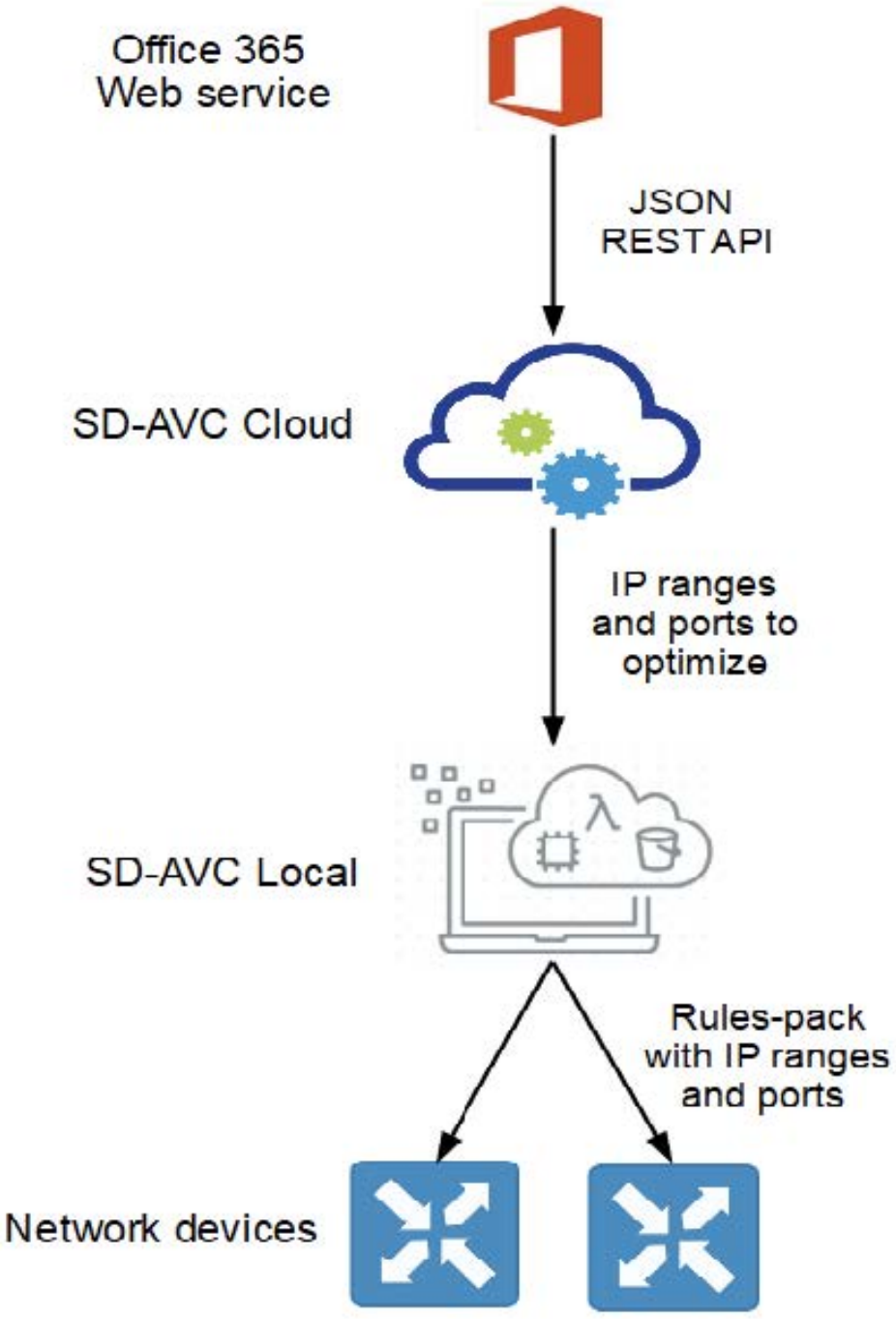


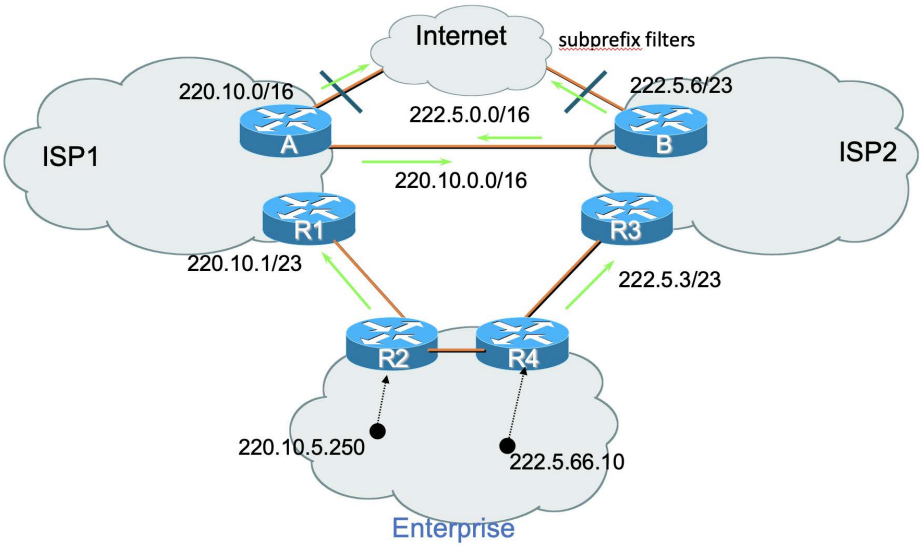
Figure 13:

Define a class map to match all the optimized category Office365 traffic, apply the ePBR policy and redirect these optimize traffic to another device, which is specified by the nexthop. You can add multiple nexthops such that if there is no route to the first nexthop, you can switch the second nexthop.

SD-AVC, which procures the traffic category information from Office365, pushes this information to specific routing devices in the network. Enhanced-PBR then matches this Office365 traffic category attribute and policy routing to a certain path, for example - bypass Firewall or local breakout.

Configure Internet Edge Load Balancing

Enterprise internet edge has multiple ISP links connected to one or multiple edge routers - R2 and R4. In order to fully utilize all the ISP links, the site-manager load balance feature is enabled on these edge devices to load



balanceinternettraffic.

Specify the LAN interfaces with “site-manager inside” and the WAN interfaces with “site-manager path”, then define the load balance policy in the site-manager primary controller to balance the load of traffic across all the WAN interfaces. You can define the DIA-class and specify the load balance method to be used - static weight, based on WAN dynamic bandwidth, and the load balance algorithm. The DIA-class is defined to specify the kind of traffic that needs to be load balanced, for example, if you require all internet traffic, you can specify this in the class so that all enterprise internal traffic is filtered by destination.

Configure Enhanced Policy-Based and Site Manager

Configure ePBR to Optimize Office 365 traffic

```
Enable
Configure terminal
class-map match-any optimize class
match traffic-category optimize
policy-map type epbr traffic-category-policy
class optimize class
set ipv4 vrf test next-hop 2.2.2.2 1.1.1.1
set ipv6 vrf test next-hop 2003::1 2002::1 2005::1
interface GigabitEthernet2
ip address 192.168.1.1 255.255.255.0
ip nbar protocol-discovery
negotiation auto
ipv6 address 2004::1/64
no mop enabled
no mop sysid
service-policy type epbr input traffic-category-policy
```

Configure Internet Edge Load Balancing

```
Enable
Configure terminal
site-manager
vrf default
master branch
source-interface Loopback0
policy local type dia
class DIA-class sequence 10
path-preference ISP1 ISP2 fallback routing
```

Border

```
site-manager
vrf default
border local
source-interface Loopback0
master 10.10.0.0
```

LAN Interface

```
interface GigabitEthernet3.30
description B1MCBR-LAN
encapsulation dot1Q 30
ip address 10.10.10.1 255.0.0.1
site-manager inside
```

WAN Interface

```
interface GigabitEthernet2.30
encapsulation dot1Q 30
ip address 10.10.10.0
site-manager path ISP1 direct-internet-access
interface GigabitEthernet3.30
encapsulation dot1Q 30
ip address 10.20.1.1
site-manager path ISP2 direct-internet-access
```

Verify the Configuration of Master traffic-classes on Primary Controller

```
Device# show site-manager master traffic-classes
Classmap: DIA-class DSCP: * [255] Traffic classid:41 classmap_id:8984
Clock Time: 17:31:04 (CST) 04/22/2020 TC Created: 00:10:48 ago
Present State: CONTROLLED
Channel1: 10 #mC30m (Gi0/0/0, LG:DIA2, BW:1000000 Kb/s, Used:752072 Kb/s, Util: 75%,
Weight:144)
Channel2: 16 #W820Z (Gi0/0/0, LG:DIA1, BW:1000000 Kb/s, Used:584202 Kb/s, Util: 58%,
Weight:255)
Load-sharing Algorithm: include-ports source destination(0x6)
Stickiness: Disabled
ICMP Probe: IP 8.8.8.8 DSCP default
Match App: No
Class-Sequence in use: 10
Class Name: DIA-class using policy best-effort
Reason for Latest Route Change: uncontrolled to Controlled Transition
Route Change History:
Date and Time Previous Exit Current Exit Reason
1:17:24:41 CST)04/22/20 None/0.0.0.0/None (Ch:0)#mC30m/3.3.3.3/Gi0/0/0Ch:10) Uncontrolled
to Controlled..
```

Verify the status of the Border Router at Branch

```
Device# show site-manager border status
Instance Status: UP
Present status last updated: 1w4d ago
Loopback: Configured Loopback0 UP (3.3.3.3)
Master: 1.1.1.1
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 00:22:12
Connection Keepalive: 10 seconds
External Wan interfaces:
Name: GigabitEthernet0/0/0 Interface Index: 8 SNMP Index: 1 SP: #mC30m Status: UP
Auto Tunnel information:
Name:Tunnel1 if_index: 24
Virtual Template: Not Configured
Borders reachable via this tunnel: 2.2.2.2
--------------------------------------------------
```

Debug Commands

- **debug site-manager master route-control**
- **debug site-manager border dia**
- **debug site-manager border route-control**
- **debug site-manager master pdp path-preference**
- **debug site-manager master pdp path-selection**

Configuring a Single Border Router

```
enable
configure terminal
 class-map match-any whitelist
    match protocol attribute application-group ms-cloud-group
    match protocol amazon-wen-services
 policy-map trype epbr SaaS-list
    class whitelist
       set ip vrf fvrf next-hop 10.20.1.1
       exit
     exit
  interface GigabitEthernet3.30
    description B1MCBR-LAN
    encapsulation dot1q 30
    ip address 10.20.0.1 255.255.255.0
    service-policy type epbr input SaaS-list
    exit

  interface GigabitEthernet2.30
    encapsulation dot1q 30
    ip vrf forwarding fvrf
    ip address 10.20.1.1. 255.255.255.0
```

Configuring Redirect for Single Border Router

```
enable
configure terminal
ip nat inside source route-map LAN interface GigabitEthernet2.30 vrf BR-LAN overload
!
interface GigabitEthernet3.30
description B1MCBR-LAN
encapsulation dot1Q 30
vrf forwarding BR-LAN
ip address 10.20.0.1 255.255.255.0
ip nbar protocol-discovery ipv4
ip nat inside
service-policy type epbr input REDIRECT
exit
!
!
interface GigabitEthernet2.30
 description B1MCBR-WAN
 encapsulation dot1q 30
 vrf forwarding fvrf
 ip address 10.20.1.1 255.255.255.0
 ip nat outside
exit
!
!
configure terminal
policy-map type epbr REDIRECT
 class AppMatchMulti
  set {ipv4 | ipv6} vrf fvrf [next-hop 10.20.1.2]
class AclMatchMulti
  set interface Dialer1
!
!
!
class-map match-all AppMatchMulti
 match protocol skype
class-map match-all AclMatchMulti
 match access-group name AclMatchMulti
end
```

Configuring Flow Stickness for Single Border Router

Use the following commands to configure flow stickness for single border router

```
enable
configure terminal
interface GigabitEthernet3.30
 description B1MCBR-LAN
 encapsulation dot1Q 30
 vrf forwarding BR-LAN
 ip address 10.20.0.1 255.255.255.0
 ip nbar protocol-discovery ipv4
 service-policy type epbr input FLOWSTICKNESS
exit
!
!
interface GigabitEthernet2.30
 description B1MCBR-WAN
 encapsulation dot1q 30
```

```
 vrf forwarding fvrf
 ip address 10.20.1.1 255.255.255.0
exit
!
!
configure terminal
 policy-map type epbr FLOWSTICKNESS
 parameter default flow-stickness
  class AppMatchMulti
   set {ipv4 | ipv6} vrf fvrf [next-hop 10.20.1.2]
  class AclMatchMulti
   set {ipv4 | ipv6} global [next-hop 10.75.1.15]
!
!
!
class-map match-all AppMatchMulti
 match protocol skype
class-map match-all AclMatchMulti
 match access-group name AclMatchMulti
end
```

Configuring Site Manager with DCA (Local Policy)

Configuration on Branch (BR1) and Master Controller (MC)

```
enable
configure terminal
    site-manager default
      vrf default
        border
          master local
        master branch
          source-interface loopback0
          policy local type dca
            class DCA sequence 1
                match application google-group policy saas-dca
                path-preference DIA1 fallback DIA2
                exit
            exit
          exit
         exit
      interface gigabitethernet3.30
        description B1MCBR-LAN
        encapsulation dot1q 30
        ip address 10.20.0.1 255.255.255.0
        site-manager inside
        exit
       exit
      interface gigabitethernet2.30
        encapsulation dot1q 30
        ip vrf forwarding fvrf
        ip address 10.20.0.1 255.255.255.0
        site-manager path DIA1 direct-internet-access
        exit
       exit
```

Configuration on Branch, BR2

```
enable
configure terminal
    site-manager default
```

```
 vrf default
   border
    source-interface loopback0
    master 192.168.3.22
   exit
  exit
 exit
interface gigabitethernet3.30
   description B1MCBR-LAN
   encapsulation dot1q 30
   ip address 10.20.0.1 255.255.255.0
   site-manager inside
   exit
  exit
 interface gigabitethernet2.30
   encapsulation dot1q 30
   ip vrf forwarding fvrf
   ip address 10.20.0.1 255.255.255.0
   site-manager path DIA2 direct-internet-access
   exit
  exit
```

Configure Site Manager with DCA (Global Policy)

Use the following commands to configure Site Manager with DCA (Global Policy). Use the following commands to configure Site Manager with DIA (Customized local Policy). If there are many branch sites requiring similar DCA policies, you can configure the policy in a central place (For example, DMVPN hub site) and the policy is published to all branch sites that have connectivity to the hub site

Configuration on Hub Master Controller

```
enable
configure terminal
    site-manager default
      vrf default
        master hub
       policy group default type DCA
        class DCA sequence 1
          match application ms-cloud-group policy saas-dca
          path-preference DIA1 fallback DIA2
        exit
       exit
      exit
```

Configuration on Branch, BR1 and Master Controller, MC

```
enable
configure terminal
    site-manager default
      vrf default
        border
          master local
        master branch
          source-interface loopback0
          hub 10.200.1.1
          exit
         exit
        exit

      interface gigabitethernet3.30
        description B1MCBR-LAN
        encapsulation dot1q 30
```

```
        ip address 10.20.0.1 255.255.255.0
        site-manager inside
        exit
       exit
      interface gigabitethernet2.30
        encapsulation dot1q 30
        ip vrf forwarding fvrf
        ip address 10.20.0.1 255.255.255.0
        site-manager path DIA1 direct-internet-access
        exit
       exit
```

Configuration on Branch, BR2

```
enable
configure terminal
    site-manager default
      vrf default
        border
         source-interface loopback0
         master 192.168.3.22
        exit
       exit
      exit
  interface gigabitethernet3.30
        description B1MCBR-LAN
        encapsulation dot1q 30
        ip address 10.20.0.1 255.255.255.0
        site-manager inside
        exit
       exit
      interface gigabitethernet2.30
        encapsulation dot1q 30
        ip vrf forwarding fvrf
        ip address 10.20.0.1 255.255.255.0
        site-manager path DIA2 direct-internet-access
        exit
       exit
```

Configure Site Manager With DIA (Local Policy)

Use the following commands to configure Site Manager with DIA (Customized local Policy). If there are many branch sites requiring similar DCA policies, you can configure the policy in a central place (For example, DMVPN hub site) and the policy is published to all branch sites that have connectivity to the hub site.

Configuration on Branch, BR1 and Master Controller, MC

```
enable
configure terminal
    ip access-list extended DIA-traffic
      deny ip 10.20.0.0 0.0.255.255
      permit ip any any
    class-map type site-manager match-any DIA-class
      match access-group DIA-traffic

    site-manager default
      vrf default
        border
          master local
        master branch
          source-interface loopback0
```

```
        policy local type DIA
         class DIA-class
         path-prefernce DIA1 fallback DIA2
        exit
       exit
      exit

    interface gigabitethernet3.30
      description B1MCBR-LAN
      encapsulation dot1q 30
      ip address 10.20.0.1 255.255.255.0
      site-manager inside
      exit
     exit
    interface gigabitethernet2.30
      encapsulation dot1q 30
      ip vrf forwarding fvrf
      ip address 10.20.0.1 255.255.255.0
      site-manager path DIA1 direct-internet-access
      exit
     exit
```

Configuration on Branch, BR2

```
enable
configure terminal
    site-manager default
      vrf default
        border
         source-interface loopback0
         master 192.168.3.22
        exit
       exit

  interface gigabitethernet3.30
        description B1MCBR-LAN
        encapsulation dot1q 30
        ip address 10.20.0.1 255.255.255.0
        site-manager inside
        exit

   interface gigabitethernet2.30
        encapsulation dot1q 30
        ip vrf forwarding fvrf
        ip address 10.20.0.1 255.255.255.0
        site-manager path DIA2 direct-internet-access
        exit
```

Configure Site Manager With DIA (Global Policy)

Use the following commands to configure Site Manager with DIA (customized global policy)

Configuration on Branch, BR1 and Master Controller, MC

```
enable
  configure terminal
    site-manager default
      vrf default
        border
          master local
```

```
        master branch
          source-interface loopback0
          hub 10.200.1.1

         exit
        exit

      interface gigabitethernet3.30
        description B1MCBR-LAN
        encapsulation dot1q 30
        ip address 10.20.0.1 255.255.255.0
        ip nat inside
        site-manager inside
        exit
       exit
      interface gigabitethernet2.30
        encapsulation dot1q 30
        ip vrf forwarding fvrf
        ip address 10.20.0.1 255.255.255.0
        ip nat outside
        site-manager path DIA1 direct-internet-access
        exit
       exit
```

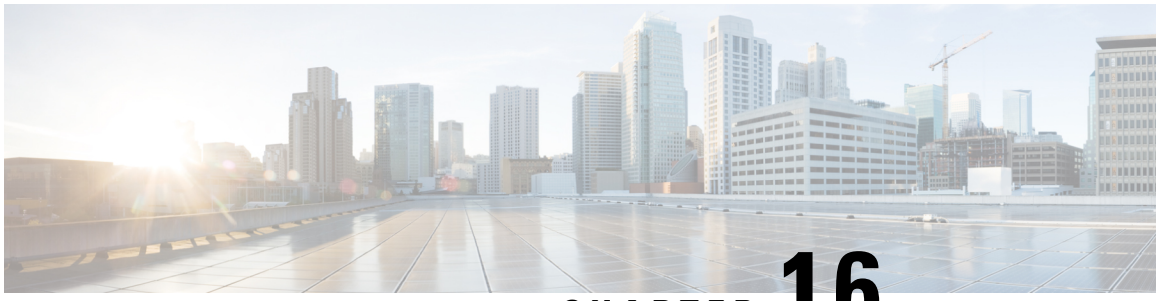
Configuration on Hub Master Controller

```
enable
configure terminal
    ip access-list extended DIA-traffic
      deny ip 10.20.0.0 0.0.255.255.
      permit ip any any
    class-map type site-manager match-any DIA-class
      match access-group DIA-traffic
   site-manager default
      vrf default
        master hub
       policy group default type DIA
        class DCA sequence 1
          match application ms-cloud-group policy saas-dca
          path-preference DIA1 fallback DIA2
        exit
       exit
      exit
```

Configuration on Branch, BR2

```
enable
configure terminal
    site-manager default
      vrf default
        border
         source-interface loopback0
         master 192.168.3.22
        exit
       exit

  interface gigabitethernet3.30
        description B1MCBR-LAN
        encapsulation dot1q 30
        ip address 10.20.0.1 255.255.255.0
        ip nat inside
        site-manager inside
```

```
     exit

interface gigabitethernet2.30
     encapsulation dot1q 30
     ip vrf forwarding fvrf
     ip address 10.20.0.1 255.255.255.0
     ip nat outside
     site-manager path DIA2 direct-internet-access
     exit
```



CHAPTER 16

PPPoE over BDI

The PPPoE over BDI feature terminates PPPoE subscribers through a VXLAN L2 overlay network onto a Cisco Bride Domain Interface (BDI).

Restrictions for PPPoE over BDI

- Service-policy queuing feature is not supported on BDI interface.
- If there is a Qos policy with queuing feature configured on the virtual template then the policy will not be applied to the session.

Information About PPPoE over BDI

PPPoE

PPPoE is a commonly used application in the deployment of digital subscriber lines (DSLs). PPPoE supports PPPoE on the client and the server.

Bridge Domain Interface

Bridge domain interface (BDI) is a logical interface that allows bidirectional flow of traffic between a Layer 2 bridged network and a Layer 3 routed network traffic. Bridge domain interfaces are identified by the same index as the bridge domain. Each bridge domain represents a Layer 2 broadcast domain. Only one bridge domain interface can be associated with a bridge domain.

Bridge domain interface supports:

- IP termination

- Layer 3 VPN termination
- Address Resolution Protocol (ARP), G-ARP, and P-ARP handling
- MAC address assignment

PPPoE over BDI

PPPoE session request from PPPoE subscriber is terminated on CSR1000v through a VxLAN tunnel. The VxLAN tunnel between Edge Router and CSR1000v provides a layer2 connection for PPPoE packets.

How to Configure PPPoE over BDI

Enabling PPPoE over BDI

```
configure terminal
interface BDI10
 no ip address
 vlan-id dot1q 10
  pppoe enable group global
exit
```

Disabling PPPoE over BDI

```
configure terminal
interface BDI10
 no ip address
 vlan-id dot1q 10
  no pppoe enable group global
exit
```

Configuration Examples for PPPoE over BDI

Configuring PPPoE over BDI

```
configure terminal
aaa new-model
aaa authentication ppp default local
username c password 0 c
!
bba-group pppoe global1
virtual-template 1
!
interface virtual-template 1
ppp ipcp address required
ip unnumbered loopback0
peer default ip address pool pool1
ppp authentication pap chap
ppp timeout retry 3
ppp timeout ncp 60
!
interface BDI10
```

```
vlan-id dot1q 10
pppoe enable group global1
!
exit
```

Additional References for PPPoE over BDI

MIBs

MIB	MIBs Link
• CISCO-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

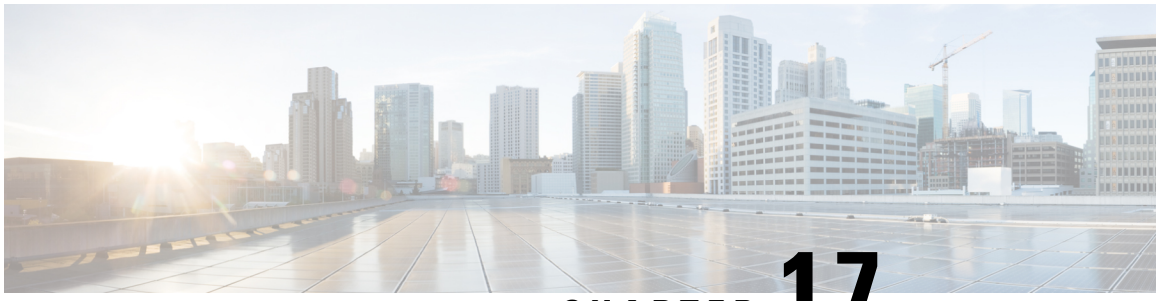
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for PPPoE over BDI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 19: Feature Information for PPPoE over BDI



CHAPTER 17

SGT Based PBR

The SGT Based PBR feature supports classification of packets based on Security Group for grouping the traffic into roles to match the defined policies in Policy-Based Routing (PBR).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Restrictions for SGT Based PBR

- SGT Based PBR feature supports policy configuration using number based tagging and does not support name based tagging.
- SGT Based PBR feature is not supported for IPV6 traffic on IOS XE.
- Dynamic route-map overrides static route-map when both are associated with the same interface. A warning message is issued during an override. The static route-map is enabled when the dynamic route-map is deleted.
- We recommend disassociating the route-map before it is deleted. You cannot configure static PBR if the route-map is deleted before disassociating it from the interface.

Information About SGT Based PBR

Cisco TrustSec

Cisco TrustSec assigns a Security Group Tag, (SGT) to the user's or device's traffic at ingress and applies the access policy based on the assigned tag. SGT Based PBR feature allows you to configure PBR based on Security Group classification enabling you to group users or devices into a role to match the defined policies.

SGT Based PBR

Security Group classification includes both Source and Destination Group, which is specified by source SGT and DGT. SGT Based PBR feature provides the PBR route-map match clause for SGT/DGT based packet classification. SGT Based PBR feature supports configuration of unlimited number of tags, but it is recommended to configure the tags based on memory available in the platform. SGT Based PBR supports VPN routing and forwarding (VRF) selection match criteria which can be used for policy based classification and forwarding of Virtual Private Network (VPN) traffic.

How to Configure SGT Based PBR

Configuring Match Security Group Tag

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag*
4. **match security-group source tag** *sgt-number*
5. **set ip next-hop** *ip-address*
6. **match security-group destination tag** *sgt-number*
7. **set ip next-hop** *ip-address*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**route-map** *map-tag* **Example:** `Device(config)# route-map policy_security`	Specifies the route-map and enters route-map configuration mode.
Step 4	**match security-group source tag** *sgt-number* **Example:** `Device(config-route-map)# match security-group source tag 100`	Configures the value for security-group source security tag.
Step 5	**set ip next-hop** *ip-address* **Example:** `Device(config-route-map)# set ip next-hop 71.71.71.6`	Specifies the next hop for routing packets.
Step 6	**match security-group destination tag** *sgt-number* **Example:** `Device(config-route-map)# match security-group destination tag 150`	Configures the value for security-group destination security tag.
Step 7	**set ip next-hop** *ip-address* **Example:** `Device(config-route-map)# set ip next-hop 72.72.72.6`	Specifies the next hop for routing packets.
Step 8	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Assigning Route-Map to an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *typeslot/ subslot/ port*[. *subinterface-number*]
4. **ip policy route-map** *map-tag*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *typeslot/ subslot/ port*[. *subinterface-number*] **Example:** `Device(config)#interface gigabitEthernet0/0/0`	Specifies the interface information and enters interface configuration mode.
Step 4	**ip policy route-map** *map-tag* **Example:** `Device(config-if)#ip policy route-map policy_security`	Assigns the route-map configured in the previous task to the interface.

Displaying and Verifying SGT Based PBR Configuration

SUMMARY STEPS

1. **enable**
2. **show ip policy**
3. **show route-map** *map-tag*
4. **show route-map dynamic**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip policy**

Example:

```
Device# show ip policy

Interface      Route map
Gi0/0/1.77     test
```

Displays IP policy information.

Step 3 **show route-map** *map-tag*

Example:

```
Device# show route-map test

route-map test, permit, sequence 10
```

```
  Match clauses:
    security-group source tag 100 111
  Set clauses:
    ip next-hop 71.71.71.6
  Policy routing matches: 0 packets, 0 bytes
route-map test, permit, sequence 20
  Match clauses:
    security-group destination tag 200 222
  Set clauses:
    ip next-hop 72.72.72.6
  Policy routing matches: 0 packets, 0 bytes
```

Displays route-map configuration.

Step 4 **show route-map dynamic**

Example:

```
Device# show route-map dynamic

route-map AAA-02/11/15-12:32:52.955-1-test, permit, sequence 0, identifier 2818572289
Match clauses:
    Security-group source tag 100 300
Set clauses:
 ip next-hop 3.3.3.2
Nexthop tracking current: 3.3.3.2
3.3.3.2, fib_nh:7FDE41661370,oce:7FDE4C540AD0,status:1

 Policy routing matches: 1012 packets, 83458 bytes
Current active dynamic routemaps = 1
```

Displays information about dynamic PBR route-map.

Configuration Examples for SGT Based PBR

Example: SGT Based PBR

The following example shows how to configure SGT Based PBR:

Example: SGT Based PBR

```
enable
 configure terminal
 route-map policy_security
 match security-group source tag 100
 match security-group source tag 111
 set ip next-hop 71.71.71.6
 match security-group destination tag 200
 match security-group destination tag 222
 set ip next-hop 72.72.72.6
 end
 interface gigabitEthernet0/0/0
 ip policy route-map policy_security
```

Additional References for SGT Based PBR

Related Documents

Related Topic	Document Title
Cisco IOS IP Routing Protocol Independent commands	Cisco IOS IP Routing Protocol Independent Command Reference
Cisco TrustSec Overview	Understanding Cisco TrustSec

Technical Assistance

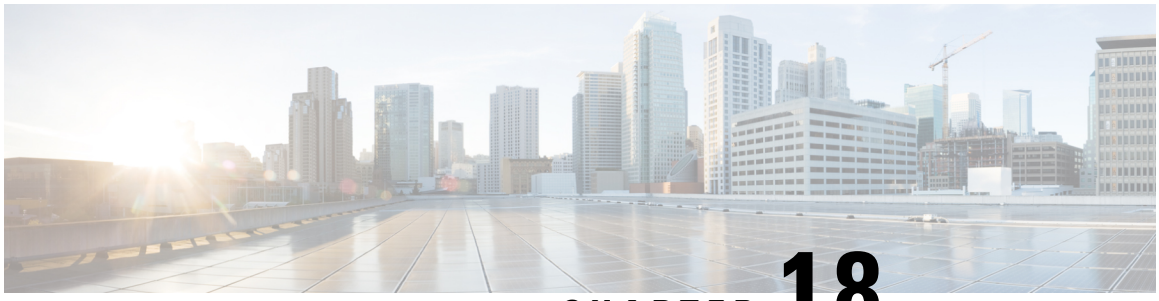
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SGT Based PBR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 20: Feature Information for SGT Based PBR



CHAPTER 18

SGT Based QoS

The SGT Based QoS feature supports the application of security group for packet classification for user group and role based or device based QoS traffic routing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SGT Based QoS

- The user groups and devices used for SGT Based QoS configuration must be assigned to the appropriate SGT groups. SGT definition and mapping can be done through Cisco ISE or through static SGT classification on the network device.

Restrictions for SGT Based QoS

- The SGT Based QoS feature does not support application prioritization within a user group.

- The SGT Based QoS feature does not support combining match application or match protocol criteria with the match sgt criteria within a policy.

Information About SGT Based QoS

SGT Based QoS

Security Group classification includes both Source and Destination Group, which is specified by source SGT and DGT. The SGT Based QoS feature enables prioritized allocation of bandwidth and QoS policies for a defined user group or device. The SGT Based QoS feature provides you the capability to assign multiple QoS policies to an application or traffic type initiated by different user groups. Each user group is defined by a unique SGT value and supports hierarchical and non-hierarchical QoS configuration. The SGT Based QoS feature supports both user group and device based QoS service levels for SGT/DGT based packet classification. The SGT Based QoS feature supports defining of user groups based on contextual information for QoS policy prioritization.

How to Configure SGT Based QoS

Configuring User Group, Device, or Role Based QoS Policies

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name*
4. **match security-group source tag** *sgt-number*
5. **match security-group destination tag** *dgt-number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* **Example:**	Specifies the class-map and enters class-map configuration mode.

	Command or Action	Purpose
	`Device(config)# class-map c1`	
Step 4	**match security-group source tag** *sgt-number* **Example:** `Device(config-cmap)# match security-group source tag 1000`	Configures the value for security-group source security tag.
Step 5	**match security-group destination tag** *dgt-number* **Example:** `Device(config-cmap)# match security-group destination tag 2000`	Configures the value for security-group destination security tag.
Step 6	**end** **Example:** `Device(config-cmap)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuring and Assigning Policy-Map to an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-map-name*
5. **bandwidth percent** *number*
6. **set dscp** *codepoint value*
7. **end**
8. **interface** *type slot/subslot/port* [. *subinterface-number*]
9. **service-policy** {**input** | **output**} *policy-map-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:**	Specifies the policy-map and enters policy-map configuration mode.

	Command or Action	Purpose
	`Device(config)# policy-map p1`	
Step 4	**class** *class-map-name* **Example:** `Device(config-pmap)# class c1`	Specifies the class and enters class configuration mode.
Step 5	**bandwidth percent** *number* **Example:** `Device(config-pmap-c)# bandwidth percent 20`	Configures the value for bandwidth percent.
Step 6	**set dscp** *codepoint value* **Example:** `Device(config-pmap-c)# set dscp ef`	Configures the Differentiated Services Code Point (DSCP) value.
Step 7	**end** **Example:** `Device(config-pmap-c)# end`	Exits policy-map class action configuration mode and returns to privileged EXEC mode.
Step 8	**interface** *type slot/subslot/port* [. *subinterface-number*] **Example:** `Device(config)#interface gigabitEthernet0/0/0.1`	Specifies the interface information and enters interface configuration mode.
Step 9	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Device(config-if)# service-policy input p1`	Assigns policy-map to the input of an interface.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Displaying and Verifying SGT Based QoS Configuration

SUMMARY STEPS

1. **enable**
2. **show class-map**
3. **debug cpl provisioning**{**api** | **db** | **errors** | **ttc**}

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show class-map**

Example:

```
Device# show class-map

 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all c1 (id 1)
   Match security-group source tag 1000
   Match security-group destination tag 2000
```

Displays class-map information.

Step 3 **debug cpl provisioning**{**api** | **db** | **errors** | **ttc**}

Example:

```
Device# debug cpl provisioning api

CPL Policy Provisioning Manager API calls debugging is on
```

Enables debugging for Call Processing Language (CPL) provisioning.

Configuration Examples for SGT Based QoS

Example: Configuring User Group, Device, or Role Based QoS Policies

The following example shows how to configure User Group, Device, or Role Based QoS Policies:

```
enable
 configure terminal
 class-map c4
  match security-group source tag 7000
  match security-group destination tag 8000
  end
 policy-map p5
  class c4
   bandwidth percent 50
   set dscp ef
   end
 interface gigabitEthernet0/0/0.1
  service-policy input p5
```

Additional References for SGT Based QoS

Related Documents

Related Topic	Document Title
Cisco IOS IP Routing Protocol Independent commands	Cisco IOS IP Routing Protocol Independent Command Reference
Cisco TrustSec Overview	Understanding Cisco TrustSec

Technical Assistance

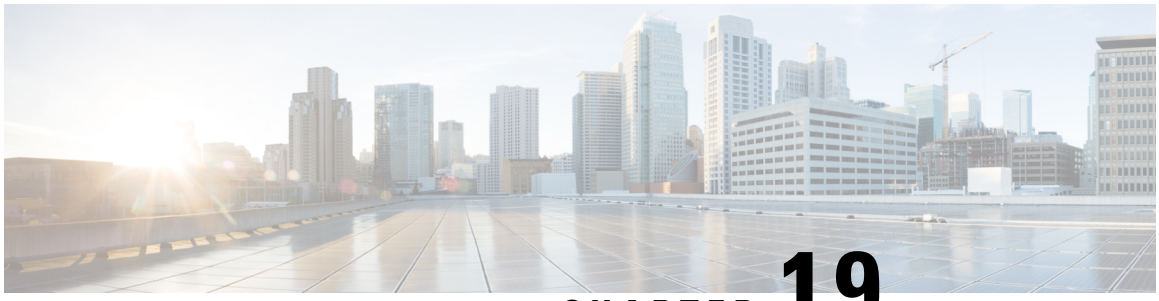
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SGT Based QoS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 21: Feature Information for SGT Based QoS



CHAPTER 19

Policy-Based Routing Default Next-Hop Routes

The Policy-Based Routing Default Next-Hop Route feature introduces the ability for packets that are forwarded as a result of the **set ip default next-hop** command to be switched at the hardware level. In prior software releases, the packets to be forwarded that are generated from the route map for policy-based routing are switched at the software level.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Policy-Based Routing Default Next-Hop Routes

Policy-Based Routing

Policy-based routing (PBR) is a process whereby the device puts packets through a route map before routing them. The route map determines which packets are routed to which device next. You might enable policy-based routing if you want certain packets to be routed some way other than the obvious shortest path. Possible applications for policy-based routing are to provide equal access, protocol-sensitive routing, source-sensitive routing, routing based on interactive versus batch traffic, and routing based on dedicated links. Policy-based routing is a more flexible mechanism for routing packets than destination routing.

To enable policy-based routing, you must identify which route map to use for policy-based routing and create the route map. The route map itself specifies the match criteria and the resulting action if all of the match clauses are met.

To enable policy-based routing on an interface, indicate which route map the device should use by using the **ip policy route-map** *map-tag* command in interface configuration mode. A packet arriving on the specified interface is subject to policy-based routing. This **ip policy route-map** command disables fast switching of all packets arriving on this interface.

To define the route map to be used for policy-based routing, use the **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq**] [*sequence-name* global configuration command.

To define the criteria by which packets are examined to learn if they will be policy-based routed, use either the **match length** *minimum-length maximum-length* command or the **match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* | *access-list-name*] command or both in route map configuration mode. No match clause in the route map indicates all packets.

To display the cache entries in the policy route cache, use the **show ip cache policy** command.

Note Mediatrace will show statistics of incorrect interfaces with policy-based routing (PBR) if the PBR does not interact with CEF or Resource Reservation Protocol (RSVP). Hence configure PBR to interact with CEF or RSVP directly so that mediatrace collects statistics only on tunnel interfaces and not physical interfaces.

Precedence Setting in the IP Header

The precedence setting in the IP header determines whether, during times of high traffic, the packets are treated with more or less precedence than other packets. By default, the Cisco software leaves this value untouched; the header remains with the precedence value that it had.

The precedence bits in the IP header can be set in the device when policy-based routing is enabled. When the packets containing those headers arrive at another device, the packets are ordered for transmission according to the precedence set, if the queueing feature is enabled. The device does not honor the precedence bits if queueing is not enabled; the packets are sent in FIFO order.

You can change the precedence setting, using either a number or name (the names came from RFC 791). You can enable other features that use the values in the **set ip precedence** route map configuration command to determine precedence. The table below lists the possible numbers and their corresponding name, from lowest to highest precedence.

Table 22: IP Precedence Values

Number	Name
0	routine
1	priority
2	immediate
3	flash
4	flash-override

Number	Name
5	critical
6	internet
7	network

The **set** commands can be used with each other. They are evaluated in the order shown in the previous table. A usable next hop implies an interface. Once the local device finds a next hop and a usable interface, it routes the packet.

How to Configure Policy-Based Routing Default Next-Hop Routes

Configuring Precedence for Policy-Based Routing Default Next-Hop Routes

Perform this task to configure the precedence of packets and specify where packets that pass the match criteria are output.

Note The **set ip next-hop** and **set ip default next-hop** commands are similar but have a different order of operation. Configuring the **set ip next-hop** command causes the system to first use policy routing and then use the routing table. Configuring the **set ip default next-hop** command causes the system to first use the routing table and then the policy-route-specified next hop.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
4. **set ip precedence** {*number* | *name*}
5. **set ip next-hop** *ip-address* [*ip-address*]
6. **set interface** *type number* [*...type number*]
7. **set ip default next-hop** *ip-address* [*ip-address*]
8. **set default interface** *type number* [*...type number*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed.
Step 4	**set ip precedence** {*number* | *name*} **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the precedence value in the IP header. **Note** You can specify either a precedence number or a precedence name.
Step 5	**set ip next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ip next-hop 192.0.2.1`	Specifies the next hop for routing packets. **Note** The next hop must be an adjacent device.
Step 6	**set interface** *type number* [*...type number*] **Example:** `Device(config-route-map)# set interface gigabitethernet 0/0/0`	Specifies the output interface for the packet.
Step 7	**set ip default next-hop** *ip-address* [*ip-address*] **Example:** `Device(config-route-map)# set ip default next-hop 172.16.6.6`	Specifies the next hop for routing packets if there is no explicit route for this destination. **Note** Like the **set ip next-hop** command, the **set ip default next-hop** command must specify an adjacent device.
Step 8	**set default interface** *type number* [*...type number*] **Example:** `Device(config-route-map)# set default interface serial 0/0/0`	Specifies the output interface for the packet if there is no explicit route for the destination.
Step 9	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuration Examples for Policy-Based Routing Default Next-Hop Routes

Example: Policy-Based Routing

The following example provides two sources with equal access to two different service providers. Packets that arrive on asynchronous interface 1/0/0 from the source 10.1.1.1 are sent to the device at 172.16.6.6 if the device has no explicit route for the destination of the packet. Packets that arrive from the source 172.17.2.2 are sent to the device at 192.168.7.7 if the device has no explicit route for the destination of the packet. All other packets for which the device has no explicit route to the destination are discarded.

```
Device(config)# access-list 1 permit ip 10.1.1.1
Device(config)# access-list 2 permit ip 172.17.2.2
Device(config)# interface async 1/0/0
Device(config-if)# ip policy route-map equal-access
Device(config-if)# exit
Device(config)# route-map equal-access permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip default next-hop 172.16.6.6
Device(config-route-map)# exit
Device(config)# route-map equal-access permit 20
Device(config-route-map)# match ip address 2
Device(config-route-map)# set ip default next-hop 192.168.7.7
Device(config-route-map)# exit
Device(config)# route-map equal-access permit 30
Device(config-route-map)# set default interface null 0
Device(config-route-map)# exit
```

Additional References

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference

Technical Assistance

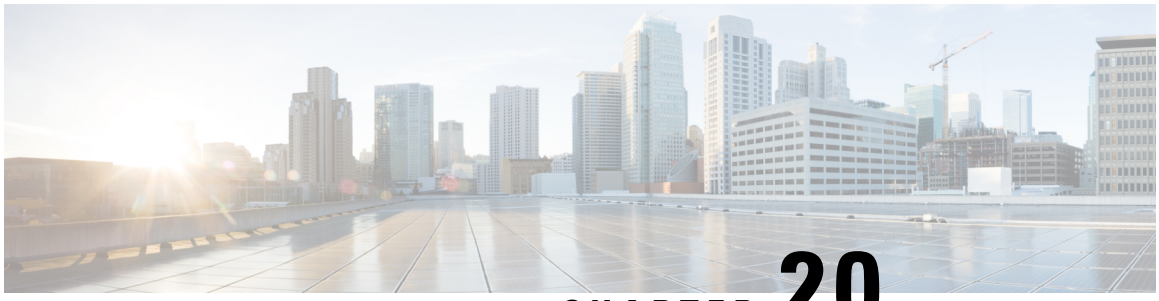
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Policy-Based Routing Default Next-Hop Routes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 23: Feature Information for Policy-Based Routing Default Next-Hop Routes



CHAPTER 20

PBR Next-Hop Verify Availability for VRF

The PBR Next-Hop Verify Availability for VRF feature enables verification of next-hop availability for IPv4/IPv6 packets in virtual routing and forwarding (VRF) instances.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About PBR Next-Hop Verify Availability for VRF

PBR Next-Hop Verify Availability for VRF Overview

Cisco IOS policy-based routing (PBR) defines packet matching and classification specifications, sets action policies, which can modify the attributes of IP packets, and overrides normal destination IP address-based routing and forwarding. PBR can be applied on global interfaces and under multiple routing instances. The PBR Next-Hop Verify Availability for VRF feature enables verification of next-hop availability for IPv4/IPv6 packets under virtual routing and forwarding (VRF) instances.

In case of an inherited VRF, the VRF instance is based on the ingress interface. Inter VRF refers to forwarding of packets from one VRF to another VRF; for example, from VRFx to VRFy. An IPv4/IPv6 packet received from VRFx is forwarded to VRFy and the availability of the next hop is verified in the VRFy instance.

How to Configure PBR Next-Hop Verify Availability for VRF

Configuring PBR Next-Hop Verify Availability for Inherited IP VRF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *vpn-route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **ip sla** *operation-number*
9. **icmp-echo** *destination-ip-address*
10. **vrf** *vrf-name*
11. **exit**
12. **ip sla schedule** *operation-number* **life forever start-time now**
13. **track** *object-number* **ip sla** *operation-number*
14. **interface** *type number*
15. **ip vrf forwarding** *vrf-name*
16. **ip address** *ip-address subnet-mask*
17. **exit**
18. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
19. **set ip vrf** *vrf-name* **next-hop verify-availability** *next-hop-address sequence* **track** *object*
20. **exit**
21. **interface** *type number*
22. **ip vrf forwarding** *vrf-name*
23. **ip policy route-map** *map-tag*
24. **ip address** *ip-address subnet-mask*
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf RED`	Configures an IP VPN routing and forwarding instance and enters VRF configuration mode.
Step 4	**rd** *vpn-route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Specifies the route distinguisher. The route distinguisher is either an autonomous system (AS) number or an IP address.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:1`	Creates a route-target extended community for a VRF and exports routing information from the target VPN extended community. The *route-target-ext-community* argument is either an AS number or an IP address.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF and imports routing information from the target VPN extended community. The *route-target-ext-community* argument is either an AS number or an IP address.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 8	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 1`	Configures a Cisco IOS IP Service Level Agreements (SLAs) operation and enters IP SLA configuration mode.
Step 9	**icmp-echo** *destination-ip-address* **Example:** `Device(config-ip-sla)# icmp-echo 10.0.0.4`	Configures an IP SLAs Internet Control Message Protocol (ICMP) echo operation and enters ICMP echo configuration mode.
Step 10	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-echo)# vrf RED`	Configures IP SLAs for a VRF instance.
Step 11	**exit** **Example:** `Device(config-ip-sla-echo)# exit`	Exits ICMP echo configuration mode and returns to global configuration mode.
Step 12	**ip sla schedule** *operation-number* **life forever start-time now** **Example:** `Device(config)# ip sla schedule 1 life forever start-time now`	Configures the scheduling parameters for a single Cisco IOS IP SLAs operation.
Step 13	**track** *object-number* **ip sla** *operation-number* **Example:**	Tracks the state of a Cisco IOS IP SLAs operation and enters tracking configuration mode.

	Command or Action	Purpose
	`Device(config)# track 1 ip sla 1`	
Step 14	**interface** *type number* **Example:** `Device(config-track)# interface Ethernet1/0`	Specifies the interface type and number and enters interface configuration mode.
Step 15	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding RED`	Configures the forwarding table.
Step 16	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.0.0.2 255.0.0.0`	Specifies the IP address and subnet mask for the interface.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. .
Step 19	**set ip vrf** *vrf-name* **next-hop verify-availability** *next-hop-address sequence* **track** *object* **Example:** `Device(config-route-map)# set ip vrf RED next-hop verify-availability 192.168.23.2 1 track 1`	Configures policy routing to verify the reachability of the next hop of a route map before the router performs policy routing to that next hop.
Step 20	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 21	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 22	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding RED`	Configures the forwarding table.
Step 23	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map test02`	Identifies a route map to use for policy routing on an interface.

	Command or Action	Purpose
Step 24	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 192.168.10.2 255.255.255.0`	Specifies the IP address and subnet mask for the interface.
Step 25	**end** **Example:** `Device(config-if)# exit`	Returns to privileged EXEC mode.

Configuring PBR Next-Hop Verify Availability for Inherited IPv6 VRF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *vpn-route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **exit**
8. **ip sla** *operation-number*
9. **icmp-echo** *destination-ip-address*
10. **vrf** *vrf-name*
11. **exit**
12. **ip sla schedule** *operation-number* **life forever start-time now**
13. **track** *object-number* **ip sla** *operation-number*
14. **interface** *type number*
15. **ip vrf forwarding** *vrf-name*
16. **ip address** *ip-address subnet-mask*
17. **ipv6 address** *ipv6-prefix*
18. **exit**
19. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
20. **set ipv6 vrf** *vrf-name* **next-hop verify-availability** *next-hop-address sequence* **track** *object*
21. **exit**
22. **interface** *type number*
23. **ip vrf forwarding** *vrf-name*
24. **ipv6 policy route-map** *map-tag*
25. **ip address** *ip-address subnet-mask*
26. **ipv6 address** *ipv6-prefix*
27. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf RED`	Configures an IP VPN routing and forwarding instance and enters VRF configuration mode.
Step 4	**rd** *vpn-route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Specifies the route distinguisher. The route distinguisher is either an autonomous system (AS) number or an IP address.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:1`	Creates a route-target extended community for a VRF and exports routing information from the target VPN extended community. The *route-target-ext-community* argument is either an AS number or an IP address.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF and imports routing information from the target VPN extended community. The *route-target-ext-community* argument is either an AS number or an IP address.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 8	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 1`	Configures a Cisco IOS IP Service Level Agreements (SLAs) operation and enters IP SLA configuration mode.
Step 9	**icmp-echo** *destination-ip-address* **Example:** `Device(config-ip-sla)# icmp-echo 10.0.0.4`	Configures an IP SLAs Internet Control Message Protocol (ICMP) echo operation and enters ICMP echo configuration mode.
Step 10	**vrf** *vrf-name* **Example:** `Device(config-ip-sla-echo)# vrf RED`	Configures IP SLAs for a VRF instance.
Step 11	**exit** **Example:**	Exits ICMP echo configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-ip-sla-echo)# exit`	
Step 12	**ip sla schedule** *operation-number* **life forever start-time now** **Example:** `Device(config)# ip sla schedule 1 life forever start-time now`	Configures the scheduling parameters for a single Cisco IOS IP SLAs operation.
Step 13	**track** *object-number* **ip sla** *operation-number* **Example:** `Device(config)# track 1 ip sla 1`	Tracks the state of a Cisco IOS IP SLAs operation and enters tracking configuration mode.
Step 14	**interface** *type number* **Example:** `Device(config-track)# interface Ethernet1/0`	Specifies the interface type and number and enters interface configuration mode.
Step 15	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding RED`	Configures the forwarding table.
Step 16	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 10.0.0.2 255.0.0.0`	Specifies the IP address and subnet mask for the interface.
Step 17	**ipv6 address** *ipv6-prefix* **Example:** `Device(config-if)# ipv6 address 2001:DB8::/48`	Specifies the IPv6 prefix.
Step 18	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 19	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. .
Step 20	**set ipv6 vrf** *vrf-name* **next-hop verify-availability** *next-hop-address sequence* **track** *object* **Example:** `Device(config-route-map)# set ipv6 vrf RED next-hop verify-availability 2001:DB8:1::1 1 track 1`	Configures policy routing to verify the reachability of the next hop of a route map before the router performs policy routing to that next hop.

	Command or Action	Purpose
Step 21	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 22	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 23	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding RED`	Configures the forwarding table.
Step 24	**ipv6 policy route-map** *map-tag* **Example:** `Device(config-if)# ipv6 policy route-map test02`	Identifies a route map to use for policy routing on an interface.
Step 25	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 192.168.10.2 255.255.255.0`	Specifies the IP address and subnet mask for the interface.
Step 26	**ipv6 address** *ipv6-prefix* **Example:** `Device(config-if)# ipv6 address 2001:DB8::/32`	Specifies the IPv6 prefix.
Step 27	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring PBR Next-Hop Verify Availability for Inter VRF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *vpn-route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **ip vrf** *vrf-name*
7. **no rd** *vpn-route-distinguisher*
8. **rd** *vpn-route-distinguisher*
9. **route-target export** *route-target-ext-community*
10. **interface** *type number*

11. **ip vrf forwarding** *vrf-name*
12. **ip address** *ip-address subnet-mask*
13. **ip policy route-map** *map-tag*
14. **interface** *type number*
15. **ip vrf forwarding** *vrf-name*
16. **ip address** *ip-address subnet-mask*
17. **exit**
18. **ip route vrf** *vrf-name prefix mask interface-type interface-number ip-address*
19. **ip route vrf** *vrf-name prefix mask ip-address*
20. Repeat Step 19 to establish additional static routes.
21. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [*sequence-name*
22. **match interface** *interface-type interface-number*
23. **set ip vrf** *vrf-name* **next-hop verify-availability** *next-hop-address sequence* **track** *object*
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf BLUE`	Configures an IP VPN routing and forwarding instance and enters VRF configuration mode.
Step 4	**rd** *vpn-route-distinguisher* **Example:** `Device(config-vrf)# rd 800:1`	Specifies the route distinguisher. The route distinguisher is either an autonomous system (AS) number or an IP address.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 800:1`	Creates a route-target extended community for a VRF and exports routing information from the target VPN extended community. The *route-target-ext-community* argument is either an AS number or an IP address.
Step 6	**ip vrf** *vrf-name* **Example:** `Device(config-vrf)# ip vrf BLUE`	Configures an IP VPN routing and forwarding instance.
Step 7	**no rd** *vpn-route-distinguisher* **Example:** `Device(config-vrf)# no rd 800:1`	Removes the specified route distinguisher.

	Command or Action	Purpose
Step 8	**rd** *vpn-route-distinguisher* **Example:** `Device(config-vrf)# rd 900:1`	Specifies the route distinguisher. The route distinguisher is either an AS number or an IP address.
Step 9	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 900:1`	Creates a route-target extended community for a VRF and exports routing information from the target VPN extended community. The *route-target-ext-community* argument is either an AS number or an IP address.
Step 10	**interface** *type number* **Example:** `Device(config-vrf)# interface Ethernet0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 11	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding RED`	Configures the forwarding table.
Step 12	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 192.168.10.2 255.255.255.0`	Specifies the IP address and subnet mask for the interface.
Step 13	**ip policy route-map** *map-tag* **Example:** `Device(config-if)# ip policy route-map test00`	Identifies a route map to use for policy routing on an interface.
Step 14	**interface** *type number* **Example:** `Device(config-if)# interface Ethernet0/1`	Specifies the interface type and number.
Step 15	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding BLUE`	Configures the forwarding table.
Step 16	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 192.168.21.1 255.255.255.0`	Specifies the IP address and subnet mask for the interface.
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 18	**ip route vrf** *vrf-name prefix mask interface-type interface-number ip-address* **Example:**	Establishes static routes.

	Command or Action	Purpose
	`Device(config)# ip route vrf BLUE 192.168.10.1 255.255.255.255 Ethernet0/0 192.168.10.1`	
Step 19	**ip route vrf** *vrf-name prefix mask ip-address* **Example:** `Device(config)# ip route vrf BLUE 192.168.23.0 255.255.255.0 192.168.21.2`	Establishes static routes.
Step 20	Repeat Step 19 to establish additional static routes.	—
Step 21	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [*sequence-name* **Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed..
Step 22	**match interface** *interface-type interface-number* **Example:** `Device(config-route-map)# match interface Ethernet0/0`	Distributes any routes that have their next hop as one of the specified interfaces.
Step 23	**set ip vrf** *vrf-name* **next-hop verify-availability** *next-hop-address sequence* **track** *object* **Example:** `Device(config-route-map)# set ip vrf BLUE next-hop verify-availability 192.168.23.2 1 track 1`	Configures policy routing to verify the reachability of the next hop of a route map of a VRF instance before the router performs policy routing to that next hop.
Step 24	**end** **Example:** `Device(config-route-map)# end`	Returns to privileged EXEC mode.

Configuration Examples for PBR Next-Hop Verify Availability for VRF

Example: Configuring PBR Next-Hop Verify Availability for Inherited IP VRF

```
Device> enable
Device# configure terminal
Device(config)# ip vrf RED
Device(config-vrf)# rd 100:1
Device(config-vrf)# route-target export 100:1
Device(config-vrf)# route-target import 100:1
Device(config-vrf)# exit
Device(config)# ip sla 1
Device(config-ip-sla)# icmp-echo 10.0.0.4
```

```
Device(config-ip-sla-echo)# vrf RED
Device(config-ip-sla-echo)# exit
Device(config)# ip sla schedule 1 life forever start-time now
Device(config)# track 1 ip sla 1
Device(config-track)# interface Ethernet0/0
Device(config-if)# ip vrf forwarding RED
Device(config-if)# ip address 10.0.0.2 255.0.0.0
Device(config-if)# exit
Device(config)# route-map test02 permit 10
Device(config-route-map)# set ip vrf RED next-hop verify-availability 192.168.23.2 1 track
 1
Device(config-route-map)# interface Ethernet0/0
Device(config-if)# ip vrf forwarding RED
Device(config-if)# ip policy route-map test02
Device(config-if)# ip address 192.168.10.2 255.255.255.0
Device(config-if)# end
```

Example: Configuring PBR Next-Hop Verify Availability for Inherited IPv6 VRF

```
Device> enable
Device# configure terminal
Device(config)# ip vrf RED
Device(config-vrf)# rd 100:1
Device(config-vrf)# route-target export 100:1
Device(config-vrf)# route-target import 100:1
Device(config-vrf)# exit
Device(config)# ip sla 1
Device(config-ip-sla)# icmp-echo 10.0.0.4
Device(config-ip-sla-echo)# vrf RED
Device(config-ip-sla-echo)# exit
Device(config)# ip sla schedule 1 life forever start-time now
Device(config)# track 1 ip sla 1
Device(config-track)# interface Ethernet0/0
Device(config-if)# ip vrf forwarding RED
Device(config-if)# ip policy route-map test02
Device(config-if)# ip address 192.168.10.2 255.255.255.0
Device(config-if)# ipv6 address 2001:DB8::/32
Device(config-if)# interface Ethernet1/0
Device(config-if)# ip vrf forwarding RED
Device(config-if)# ip address 10.0.0.2 255.0.0.0
Device(config-if)# ipv6 address 2001:DB8::/48
Device(config-if)# exit
Device(config)# route-map test02 permit 10
Device(config-route-map)# set ipv6 vrf RED next-hop verify-availability 2001:DB8:1::1 1
track 1
Device(config-route-map)# end
```

Example: Configuring PBR Next-Hop Verify Availability for Inter VRF

```
Device> enable
Device# configure terminal
Device(config)# ip vrf BLUE
Device(config-vrf)# rd 800:1
Device(config-vrf)# route-target export 800:1
Device(config-vrf)# ip vrf BLUE
Device(config-vrf)# no rd 800:1
Device(config-vrf)# rd 900:1
```

```
Device(config-vrf)# route-target export 900:1
Device(config-vrf)# interface Ethernet0/0
Device(config-if)# ip vrf forwarding RED
Device(config-if)# ip address 192.168.10.2 255.255.255.0
Device(config-if)# ip policy route-map test00
Device(config-if)# interface Ethernet0/1
Device(config-if)# ip vrf forwarding BLUE
Device(config-if)# ip address 192.168.21.1 255.255.255.0
Device(config-if)# exit
Device(config)# ip route vrf blue 192.168.10.1 255.255.255.255 Ethernet0/0 192.168.10.1
Device(config)# ip route vrf blue 192.168.23.0 255.255.255.0 192.168.21.2
Device(config)# route-map test00 permit 10
Device(config-route-map)# match interface Ethernet0/0
Device(config-route-map)# set ip vrf blue next-hop verify-availability 192.168.23.2 1 track
 1
Device(config-route-map)# end
```

Additional References for PBR Next-Hop Verify Availability for VRF

Related Documents

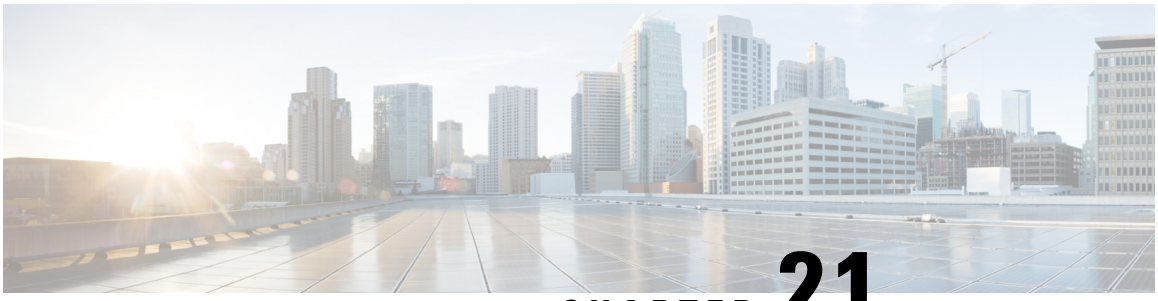
Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for PBR Next-Hop Verify Availability for VRF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 21

QoS Policy Propagation via BGP

The QoS Policy Propagation via BGP feature allows you to classify packets by IP precedence based on the Border Gateway Protocol (BGP) community lists, BGP autonomous system paths, and access lists. After packets have been classified, you can use other quality of service (QoS) features such as committed access rate (CAR) and Weighted Random Early Detection (WRED) to specify and enforce policies to fit your business model.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for QoS Policy Propagation via BGP

- Enable the Border Gateway Protocol (BGP) and Cisco Express Forwarding (CEF) or distributed CEF (dCEF) on the device. Subinterfaces on an ATM interface that have the **bgp-policy** command enabled must use CEF mode because dCEF is not supported. dCEF uses the Versatile Interface Processor (VIP) rather than the Route Switch Processor (RSP) to perform forwarding functions.
- Define the policy.
- Apply the policy through BGP.

- Configure the BGP community list, BGP autonomous system path, or access list and enable the policy on an interface.
- Enable committed access rate (CAR) or Weighted Random Early Detection (WRED) to use the policy.

Information About QoS Policy Propagation via BGP

Benefits of QoS Policy Propagation via BGP

The QoS Policy Propagation via BGP feature allows you to classify packets by IP precedence based on Border Gateway Protocol (BGP) community lists, BGP autonomous system paths, and access lists. After a packet has been classified, you can use other quality of service (QoS) features such as committed access rate (CAR) and Weighted Random Early Detection (WRED) to specify and enforce policies to fit your business model.

How to Configure QoS Policy Propagation via BGP

Configuring QoS Policy Propagation via BGP Based on Community Lists

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [
4. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
5. **set ip precedence** [*number* | *name*]
6. **exit**
7. **router bgp** *autonomous-system*
8. **table-map** *route-map-name*
9. **exit**
10. **ip community-list** *standard-list-number* {**permit** | **deny**} [*community-number*]
11. **interface** *type number*
12. **bgp-policy** {**source** | **destination**} **ip-prec-map**
13. **exit**
14. **ip bgp-community new-format**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**Example:** `Device(config)# route-map alpha permit ordering-seq`	Configures a route map and specifies how the packets are to be distributed. .
Step 4	**match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]} **Example:** `Device(config-route-map)# match community 1`	Matches a Border Gateway Protocol (BGP) community list.
Step 5	**set ip precedence** [*number* | *name*] **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the community list matches. **Note** You can specify either a precedence number or a precedence name.
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 45000`	Enables a BGP process and enters router configuration mode.
Step 8	**table-map** *route-map-name* **Example:** `Device(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 9	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 10	**ip community-list** *standard-list-number* {**permit** | **deny**} [*community-number*] **Example:**	Creates a community list for BGP and controls access to it.

	Command or Action	Purpose
	`Device(config)# ip community-list 1 permit 2`	
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 12	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Device(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 14	**ip bgp-community new-format** **Example:** `Device(config)# ip bgp-community new-format`	(Optional) Displays the BGP community number in AA:NN (autonomous system:community number/4-byte number) format.
Step 15	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation via BGP Based on the Autonomous System Path Attribute

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **named-ordering-route-map enable**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq** *sequence-name*
5. **match as-path** *path-list-number*
6. **set ip precedence** [*number* | *name*]
7. **exit**
8. **router bgp** *autonomous-system*
9. **table-map** *route-map-name*
10. **exit**
11. **ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression*
12. **interface** *type number*

13. **bgp-policy** {**source** | **destination**} **ip-prec-map**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**named-ordering-route-map enable**] **Example:** `Device(config)# named-ordering-route-map enable`	Enables ordering of route-maps based on a string provided by the user.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq** *sequence-name* **Example:** `Device(config)# route-map alpha permit ordering-seq sequence1`	Configures a route map and specifies how the packets are to be distributed. **ordering-seq** indicates the sequence that is to be used for ordering of route-maps.
Step 5	**match as-path** *path-list-number* **Example:** `Device(config-route-map)# match as-path 2`	Matches a Border Gateway Protocol (BGP) autonomous system path access list.
Step 6	**set ip precedence** [*number* | *name*] **Example:** `Device(config-route-map)# set ip precedence 5`	Sets the IP Precedence field when the autonomous-system path matches. **Note** You can specify either a precedence number or a precedence name.
Step 7	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 8	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 45000`	Enables a BGP process and enters router configuration mode.
Step 9	**table-map** *route-map-name* **Example:**	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.

	Command or Action	Purpose
	`Device(config-router)# table-map rm1`	
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 11	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expression* **Example:** `Device(config)# ip as-path access-list 500 permit 45000`	Defines an autonomous system path access list.
Step 12	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interface (or subinterface) and enters interface configuration mode.
Step 13	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Device(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP precedence.
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring QoS Policy Propagation via BGP Based on an Access List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **named-ordering-route-map enable**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq** *sequence-name*
5. **match ip address** *access-list-number*
6. **set ip precedence** [*number* | *name*]
7. **exit**
8. **router bgp** *autonomous-system*
9. **table-map** *route-map-name*
10. **exit**
11. **access-list** *access-list-number* {**permit** | **deny**} *source*
12. **interface** *type number*
13. **bgp-policy** {**source** | **destination**} **ip-prec-map**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**named-ordering-route-map enable**] **Example:** `Device(config)# named-ordering-route-map enable`	Enables ordering of route-maps based on a string provided by the user.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] [**ordering-seq** *sequence-name* **Example:** `Device(config)# route-map alpha permit ordering-seq sequence1`	Configures a route map and specifies how the packets are to be distributed. **ordering-seq** indicates the sequence that is to be used for ordering of route-maps.
Step 5	**match ip address** *access-list-number* **Example:** `Device(config-route-map)# match ip address 69`	Matches an access list.
Step 6	**set ip precedence** [*number* | *name*] **Example:** `Device(config-route-map)# set ip precedence routine`	Sets the IP precedence field when the autonomous system path matches.
Step 7	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 8	**router bgp** *autonomous-system* **Example:** `Device(config)# router bgp 45000`	Enables a Border Gateway Protocol (BGP) process and enters router configuration mode.
Step 9	**table-map** *route-map-name* **Example:** `Device(config-router)# table-map rm1`	Modifies the metric and tag values when the IP routing table is updated with BGP learned routes.
Step 10	**exit** **Example:**	Exits router configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-router)# exit`	
Step 11	**access-list** *access-list-number* {**permit** | **deny**} *source* **Example:** `Device(config)# access-list 69 permit 10.69.0.0`	Defines an access list.
Step 12	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies the interfaces (or subinterface) and enters interface configuration mode.
Step 13	**bgp-policy** {**source** | **destination**} **ip-prec-map** **Example:** `Device(config-if)# bgp-policy source ip-prec-map`	Classifies packets using IP Precedence.
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring QoS Policy Propagation via BGP

To monitor the QoS Policy Propagation via the BGP feature configuration, use the following optional commands.

Command or Action	Purpose
show ip bgp	Displays entries in the Border Gateway Protocol (BGP) routing table to verify whether the correct community is set on the prefixes.
show ip bgp community-list *community-list-number*	Displays routes permitted by the BGP community to verify whether correct prefixes are selected.
show ip cef *network*	Displays entries in the forwarding information base (FIB) table based on the specified IP address to verify whether Cisco Express Forwarding has the correct precedence value for the prefix.
show ip interface	Displays information about the interface.

Command or Action	Purpose
show ip route *prefix*	Displays the current status of the routing table to verify whether correct precedence values are set on the prefixes.

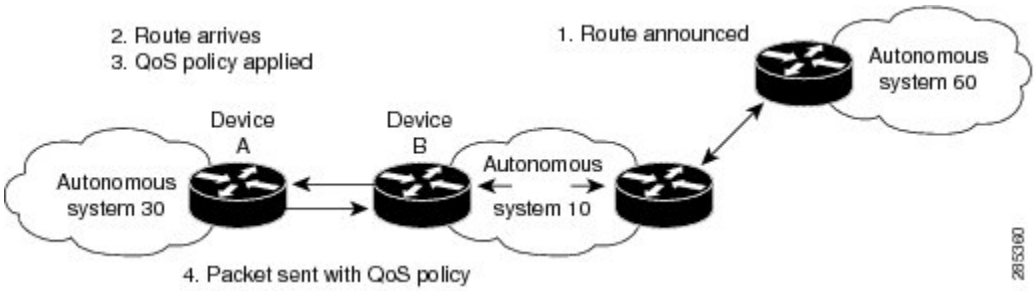
Configuration Examples for QoS Policy Propagation via BGP

Example: Configuring QoS Policy Propagation via BGP

The following example shows how to create route maps to match access lists, Border Gateway Protocol (BGP) community lists, and BGP autonomous system paths, and apply IP precedence to routes learned from neighbors.

In the figure below, Device A learns routes from autonomous system 10 and autonomous system 60. The quality of service (QoS) policy is applied to all packets that match defined route maps. Any packets from Device A to autonomous system 10 or autonomous system 60 are sent the appropriate QoS policy, as the numbered steps in the figure indicate.

Figure 14: Device Learning Routes and Applying QoS Policy



Device A Configuration

```
interface serial 5/0/0/1:0
ip address 10.28.38.2 255.255.255.0
bgp-policy destination ip-prec-map
no ip mroute-cache
no cdp enable
frame-relay interface-dlci 20 IETF
router bgp 30
 table-map precedence-map
 neighbor 10.20.20.1 remote-as 10
 neighbor 10.20.20.1 send-community
!
ip bgp-community new-format
!
! Match community 1 and set the IP precedence to priority
route-map precedence-map permit 10
 match community 1
 set ip precedence priority
!
! Match community 2 and set the IP precedence to immediate
route-map precedence-map permit 20
```

```
 match community 2
 set ip precedence immediate
!
! Match community 3 and set the IP precedence to flash
route-map precedence-map permit 30
 match community 3
 set ip precedence flash
!
! Match community 4 and set the IP precedence to flash-override
route-map precedence-map permit 40
 match community 4
 set ip precedence flash-override
!
! Match community 5 and set the IP precedence to critical
route-map precedence-map permit 50
 match community 5
 set ip precedence critical
!
! Match community 6 and set the IP precedence to internet
route-map precedence-map permit 60
 match community 6
 set ip precedence internet
!
! Match community 7 and set the IP precedence to network
route-map precedence-map permit 70
 match community 7
 set ip precedence network
!
! Match ip address access list 69 or match autonomous system path 1
! and set the IP precedence to critical
route-map precedence-map permit 75
 match ip address 69
 match as-path 1
 set ip precedence critical
!
! For everything else, set the IP precedence to routine
route-map precedence-map permit 80
 set ip precedence routine
!
! Define community lists
ip community-list 1 permit 60:1
ip community-list 2 permit 60:2
ip community-list 3 permit 60:3
ip community-list 4 permit 60:4
ip community-list 5 permit 60:5
ip community-list 6 permit 60:6
ip community-list 7 permit 60:7
!
! Define the AS path
ip as-path access-list 1 permit ^10_60
!
! Define the access list
access-list 69 permit 10.69.0.0
```

Device B Configuration

```
router bgp 10
 neighbor 10.30.30.1 remote-as 30
 neighbor 10.30.30.1 send-community
 neighbor 10.30.30.1 route-map send_community out
!
ip bgp-community new-format
!
```

```
! Match prefix 10 and set community to 60:1
route-map send_community permit 10
 match ip address 10
 set community 60:1
!
! Match prefix 20 and set community to 60:2
route-map send_community permit 20
 match ip address 20
 set community 60:2
!
! Match prefix 30 and set community to 60:3
route-map send_community permit 30
 match ip address 30
 set community 60:3
!
! Match prefix 40 and set community to 60:4
route-map send_community permit 40
 match ip address 40
 set community 60:4
!
! Match prefix 50 and set community to 60:5
route-map send_community permit 50
 match ip address 50
 set community 60:5
!
! Match prefix 60 and set community to 60:6
route-map send_community permit 60
 match ip address 60
 set community 60:6
!
! Match prefix 70 and set community to 60:7
route-map send_community permit 70
 match ip address 70
 set community 60:7
!
! For all others, set community to 60:8
route-map send_community permit 80
 set community 60:8
!
! Define access lists
access-list 10 permit 10.61.0.0
access-list 20 permit 10.62.0.0
access-list 30 permit 10.63.0.0
access-list 40 permit 10.64.0.0
access-list 50 permit 10.65.0.0
access-list 60 permit 10.66.0.0
access-list 70 permit 10.67.0.0
```

Additional References

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference
BGP configuration	*BGP Configuration Guide*

Related Topic	Document Title
Cisco Express Forwarding configuration	*Cisco Express Forwarding Configuration Guide*
Committed access rate configuration	"Configuring Committed Access Rate" module in the *QoS: Classification Configuration Guide* (part of the Quality of Service Solutions Configuration Guide Library)
Weighted Random Early Detection configuration	"Configuring Weighted Random Early Detection" module in the *QoS: Congestion Avoidance Configuration Guide* (part of the Quality of Service Solutions Configuration Guide Library)

Technical Assistance

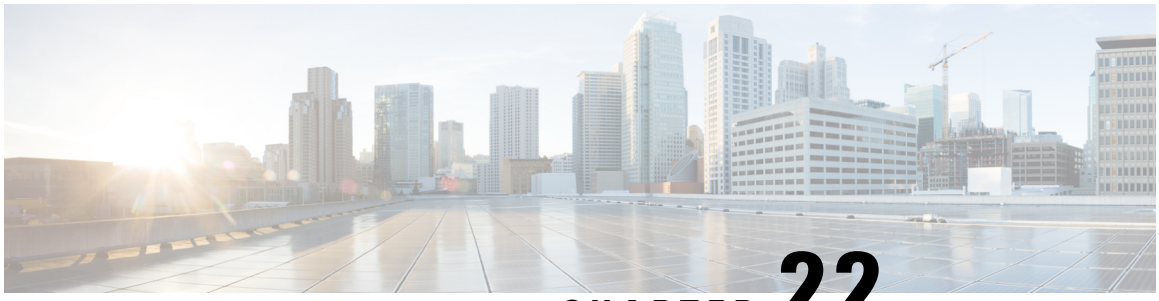
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for QoS Policy Propagation via BGP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 24: Feature Information for QoS Policy Propagation via BGP



CHAPTER 22

NetFlow Policy Routing

NetFlow policy routing (NPR) integrates policy routing, which enables traffic engineering and traffic classification, with NetFlow services, which provide billing, capacity planning, and information monitoring on real-time traffic flows. IP policy routing works with Cisco Express Forwarding (formerly known as CEF), distributed Cisco Express Forwarding (formerly known as dCEF), and NetFlow.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for NetFlow Policy Routing

For NetFlow policy routing to work, the following features must already be configured:

- Cisco Express Forwarding, distributed Cisco Express Forwarding, or NetFlow
- Policy routing

Restrictions for NetFlow Policy Routing

- NetFlow Policy Routing (NPR) is available only on Cisco platforms that support Cisco Express Forwarding.

- Distributed Forwarding Information Base (FIB)-based policy routing is available only on platforms that support distributed Cisco Express Forwarding.
- The **set ip next-hop verify-availability** command is not supported in distributed Cisco Express Forwarding because distributed Cisco Express Forwarding does not support the Cisco Discovery Protocol (formerly known as CDP) database.

Information About NetFlow Policy Routing

NetFlow Policy Routing

NetFlow policy routing (NPR) integrates policy routing, which enables traffic engineering and traffic classification, with NetFlow services, which provide billing, capacity planning, and information monitoring on real-time traffic flows. IP policy routing works with Cisco Express Forwarding (formerly known as CEF), distributed Cisco Express Forwarding (formerly known as dCEF), and NetFlow.

NetFlow policy routing leverages the following technologies:

- Cisco Express Forwarding, which looks at a Forwarding Information Base (FIB) instead of a routing table when switching packets, to address maintenance problems of a demand caching scheme.
- Distributed Cisco Express Forwarding, which addresses the scalability and maintenance problems of a demand caching scheme.
- NetFlow, which provides accounting, capacity planning, and traffic monitoring capabilities.

The following are the benefits of NPR:

- NPR takes advantage of new switching services. Cisco Express Forwarding, distributed Cisco Express Forwarding, and NetFlow can now use policy routing.
- Policy routing can be deployed on a wide scale and on high-speed interfaces.

NPR is the default policy routing mode. No additional configuration tasks are required to enable policy routing with Cisco Express Forwarding, distributed Cisco Express Forwarding, or NetFlow. As soon as one of these features is turned on, packets are automatically subjected to policy routing in the appropriate switching path.

The following example shows how to configure policy routing with Cisco Express Forwarding. The route is configured to verify that the next hop 10.0.0.8 of the route map named test is a Cisco Discovery Protocol neighbor before the device tries to policy-route to it.

```
Device(config)# ip cef
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip route-cache flow
Device(config-if)# ip policy route-map test
Device(config-if)# exit
Device(config)# route-map test permit 10
Device(config-route-map)# match ip address 1
Device(config-route-map)# set ip precedence priority
Device(config-route-map)# set ip next-hop 10.0.0.8
Device(config-route-map)# set ip next-hop verify-availability
Device(config-route-map)# exit
Device(config)# route-map test permit 20
Device(config-route-map)# match ip address 101
```

```
Device(config-route-map)# set interface Ethernet 0/0/3
Device(config-route-map)# set ip tos max-throughput
Device(config-route-map)# exit
```

Next-Hop Reachability

You can use the **set ip next-hop verify-availability** command to configure policy routing to verify the reachability of the next hop of a route map before the device performs policy routing to that next hop. This command has the following restrictions:

- It can cause performance degradation.
- Cisco Discovery Protocol must be enabled on the interface.
- The directly connected next hop must be a Cisco Discovery Protocol-enabled Cisco device.
- It does not work with distributed Cisco Express Forwarding configurations.

If a device is policy routing packets to the next hop and the next hop happens to be down, the device tries unsuccessfully to use the Address Resolution Protocol (ARP). This behavior can continue indefinitely. You can prevent this behavior by configuring the **set ip next-hop verify availability** command on the device. This command first verifies (using a route map) whether the next hop is a Cisco Discovery Protocol neighbor of the device before routing packets to that next hop. However, if you configure this command on a device whose next hop is not a Cisco Discovery Protocol neighbor, the device looks at the subsequent next hop, if there is one. If there is no available next hop, packets are not policy-routed. This configuration is optional because some media or encapsulations do not support Cisco Discovery Protocol.

If the **set ip next-hop verify availability** command is not configured, packets are either policy-routed or remain forever unrouted.

If you want to verify the availability of only some next hops, you can configure different route-map entries (under the same route-map name) with different criteria (using access-list matching or packet-size matching), and use the **set ip next-hop verify availability** configuration command selectively.

Additional References

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference

Technical Assistance

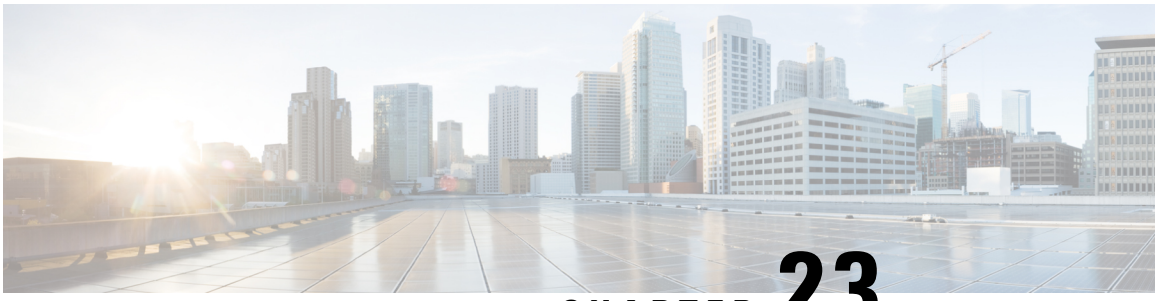
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NetFlow Policy Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 25: Feature Information for NetFlow Policy Routing



CHAPTER 23

Recursive Static Route

The Recursive Static Route feature enables you to install a recursive static route into the Routing Information Base (RIB) even if the next-hop address of the static route or the destination network itself is already available in the RIB as part of a previously learned route. This module explains recursive static routes and how to configure the Recursive Static Route feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Recursive Static Route

When recursive static routes are enabled using route maps, only one route map can be entered per virtual routing and forwarding (VRF) instance or topology. If a second route map is entered, the new map will overwrite the previous one.

Information About Recursive Static Route

How to Install Recursive Static Route

Installing Recursive Static Routes in a VRF

Perform these steps to install recursive static routes in a specific virtual routing and forwarding (VRF) instance. You can configure the recursive-static-route functionality on any number of VRFs. Installing recursive static routes in specific VRFs allows you to retain the default RIB behavior (of removing recursive static routes) for the rest of the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4 | ipv6**}
6. **exit**
7. **exit**
8. **ip route** [**vrf** *vrf-name*] *prefix mask ip-address*
9. **ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*]
10. **end**
11. **show running-config | include install**
12. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Creates a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:**	Specifies a route distinguisher for a VRF instance.

	Command or Action	Purpose
	`Device(config-vrf)# rd 100:1`	
Step 5	**address-family** {**ipv4** | **ipv6**} **Example:** `Device(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to specify an IPv4 or IPv6 address family for a VRF.
Step 6	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits VRF address family configuration mode.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode.
Step 8	**ip route** [**vrf** *vrf-name*] *prefix mask ip-address* **Example:** `Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1`	Configures a static route for a specific VRF instance.
Step 9	**ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*] **Example:** `Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1`	Enables recursive static routes to be installed in the RIB of a specific VRF instance.
Step 10	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**show running-config | include install** **Example:** `Device# show running-config | inc install`	Displays all recursive static route configurations.
Step 12	**show ip route vrf** *vrf-name* **Example:** `Device# show ip route vrf vrf1`	Displays the IP routing table associated with a specific VRF.

Installing Recursive Static Routes Using a Route Map

Perform this task to install recursive static routes in a virtual routing and forwarding (VRF) instance defined by a route map. You can perform this task if you want to install recursive static routes for only a certain range of networks. If the **route-map** keyword is used without the **vrf** keyword, recursive static routes defined by the route map will be applicable for the global VRF or topology.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4 | ipv6**}
6. **exit**
7. **exit**
8. **ip route** [**vrf** *vrf-name*] *prefix mask ip-address*
9. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
10. **route-map** *map-tag*
11. **match ip address** *access-list-number*
12. **exit**
13. **ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*] [**route-map** *map-name*]
14. **end**
15. **show running-config | include install**
16. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Creates a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Specifies a route distinguisher for a VRF instance.
Step 5	**address-family** {**ipv4	ipv6**} **Example:** `Device(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to specify an IPv4 or an IPv6 address-family type for a VRF.
Step 6	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits VRF address family configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode.
Step 8	**ip route** [**vrf** *vrf-name*] *prefix mask ip-address* **Example:** `Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1`	Configures a static route for a specific VRF instance.
Step 9	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 10 permit 10.0.2.0 255.255.255.0`	Defines a standard access list permitting addresses that need to be translated.
Step 10	**route-map** *map-tag* **Example:** `Device(config)# route-map map1`	Defines a route map to control route redistribution and enters route-map configuration mode.
Step 11	**match ip address** *access-list-number* **Example:** `Device(config-route-map)# match ip address 10`	Matches routes that have a destination network address that is permitted by a standard or extended access list.
Step 12	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode.
Step 13	**ip route static install-routes-recurse-via-nexthop** [**vrf** *vrf-name*] [**route-map** *map-name*] **Example:** `Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1 route-map map1`	Enables installation of recursive static routes defined by a route map into the RIB of a specific VRF.
Step 14	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 15	**show running-config | include install** **Example:** `Device# show running-config | inc install`	Displays all recursive static route configurations.
Step 16	**show ip route vrf** *vrf-name* **Example:** `Device# show ip route vrf vrf1`	Displays the IP routing table associated with a specific VRF.

Configuration Examples for Recursive Static Route

Example: Installing Recursive Static Routes in a VRF

The following example shows how to install recursive static routes into a specific virtual routing and forwarding instance. By using the **vrf** keyword, you can ensure that recursive static routes are installed in the Routing Information Base (RIB) of only the specified VRF. The rest of the network retains the default behavior of not installing recursive static routes in the RIB. This example is based on the assumption that a 10.0.0.0/8 route is already installed dynamically or statically in the RIB of vrf1.

```
Device> enable
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# rd 1:100
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1
Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1
Device(config)# end
```

Example: Installing Recursive Static Routes using a Route Map

You can use the **route-map** keyword to install recursive static routes defined by the route map into the Routing Information Base (RIB). You can also specify a route map for a specific virtual routing and forwarding (VRF) instance to ensure that the route map is applied to only the specified VRF. In the example given below, a route map is specified for a specific VRF. This example is based on the assumption that a 10.0.0.0/8 route is already installed statically or dynamically in the RIB of vrf1.

```
Device> enable
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# rd 100:2
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# access-list 10 permit 10.0.2.0 255.255.255.0
Device(config)# route-map map1
Device(config-route-map)# match ip address 10
Device(config-route-map)# exit
Device(config)# ip route static install-routes-recurse-via-nexthop vrf vrf1 route-map map1
Device(config)# ip route vrf vrf1 10.0.2.0 255.255.255.0 10.0.1.1
Device(config)# ip route vrf vrf1 10.0.3.0 255.255.255.0 10.0.1.1
Device(config)# end
```

In the example above, route 10.0.2.0 255.255.255.0 10.0.1.1 will be installed in the RIB, but the route 10.0.3.0 255. 255.255.0 10.0.1.1 will not be installed in the RIB because this route does not match the network defined in the route map.

Additional References for Recursive Static Route

Related Documents

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference

Technical Assistance

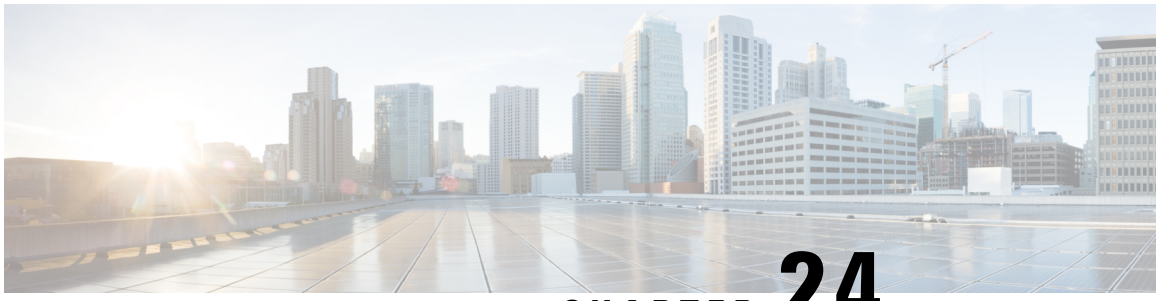
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Recursive Static Routes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 26: Feature Information for Recursive Static Routes



CHAPTER 24

TCP Authentication Option

With TCP Authentication Option (TCP-AO), defined in RFC 5925, you can protect long-lived TCP connections against replays using stronger Message Authentication Codes (MACs).

Overview of TCP Authentication Option

TCP-AO is the proposed replacement for TCP MD5, defined in RFC 2385. Unlike TCP MD5, TCP-AO is resistant to collision attacks and provides algorithmic agility and support for key management.

TCP-AO has the following distinct features:

- TCP-AO supports the use of stronger Message Authentication Codes (MACs) to enhance the security of long-lived TCP connections.
- TCP-AO protects against replays for long-lived TCP connections, and coordinates key changes between endpoints by providing a more explicit key management.

TCP-AO is supported along with TCP MD5, and you can choose one of the authentication methods. However, a configuration in which one of the devices is configured with the TCP MD5 option and the other with the TCP-AO option is not supported.

TCP-AO Key Chain

TCP-AO is based on traffic keys and Message Authentication Codes (MACs) generated using the keys and a MAC algorithm. The traffic keys are derived from primary keys that you can configure in a TCP-AO key chain. Use the **key chain** *key-chain-name* **tcp** command in the global configuration mode to create a TCP-AO key chain and configure keys in the chain. The TCP-AO key chain must be configured on both the peers communicating via a TCP connection.

Keys in a TCP-AO key chain have the following configurable properties:

Configurable Property	Description
send-id	Key identifier of the TCP-AO option of the outgoing segment. The send identifier configured on a router must match the receive identifier configured on the peer.
recv-id	Key identifier compared with the TCP-AO key identifier of the incoming segment during authentication. The receive identifier configured on a router must match the send identifier configured on the peer.
cryptographic-algorithm	The MAC algorithm to be used to create MACs for outgoing segments. The algorithm can be one of the following: • AES-128-CMAC authentication algorithm • HMAC-SHA-1 authentication algorithm • HMAC-SHA-256 authentication algorithm.
include-tcp-options	This flag indicates whether TCP options other than TCP-AO will be used to calculate MACs. With this flag enabled, the contents of all options along with a zero-filled authentication option, is used to calculate the MAC. When the flag is disabled, all options other than TCP-AO are excluded from MAC calculations. This flag is disabled by default. **Note** The configuration of this flag is overridden by the application configuration when the application configuration is available.
send-lifetime	This configuration determines the time for which a key is valid and can be used for TCP-AO-based authentication of TCP segments to be sent. When the lifetime of key elapses and the key expires, the next key with the longest lifetime is selected.
accept-lifetime	This configuration determines the time for which a key is valid and can be used for TCP-AO-based authentication of received TCP segments.
key-string	The key string is a pre-shared primary key configured on both peers and is used to derive the traffic keys.

Configurable Property	Description
accept-ao-mismatch	This flag determines whether the receiver accepts segments for which the MAC in the incoming TCP-AO does not match the MAC generated on the receiver. With this configuration, incoming segments without TCP Authentication Option are also accepted. **Note** • Use this configuration with caution. This configuration disables TCP-AO functionality and key rollover on associated connections. • The configuration of this flag is overridden by the application configuration when the application configuration is available.

Primary Key Tuples

The key chain and keys are used to create Primary Key Tuples that are optimized for look-ups during TCP send and receive operations. The Primary Key Tuples consists of a primary key, identifiers for the key, algorithms to be used for the Key Derivation Function (KDF) and MAC, and other properties.

On both the peers, two pointers called current-key and next-key are used to track Primary Key Tuples .

- current-key: Identifies the Primary Key Tuples that is being used to compute traffic keys for outgoing TCP segments.
- next-key: Identifies the Primary Key Tuples that is ready to be used to authenticate received segments.

Traffic Keys

Traffic keys are used to compute MACs of segment data using an MAC algorithm. Traffic keys are derived using a Key Derivation Function (KDF) from an Primary Key Tuples and the KDF context. The KDF context consists of the local and remote IP address pairs and TCP port numbers. For established connections, the KDF context also includes the TCP Initial Sequence Numbers (ISNs) in each direction.

A single Primary Key Tuple can be used to derive the four traffic keys in the following list. An endpoint uses at least three of the keys for authentication.

- Send SYN Traffic Key – the traffic key used to authenticate outgoing SYNs.
- Receive SYN Traffic Key – the traffic key used to authenticate incoming SYNs.
- Send Other Key – the traffic key used to authenticate all other outgoing TCP segments.
- Receive Other Key – the traffic key used to authenticate all other incoming TCP segments.

Message Authentication Codes

An MAC is computed for a TCP segment using the configured MAC algorithm, relevant traffic keys, and the TCP segment data prefixed with a pseudo-header.

Protection from Replays in Long-lived TCP Connections

The 32-bit sequence number of TCP segments may roll over and repeat in the case of long-lived TCP connections. As a result of a repetition of sequence numbers, TCP Segments may get replayed within a

connection. To avoid this, TCP-AO uses a 32-bit Sequence Number Extension (SNE) in the pseudo-header along with the TCP sequence number for transmitted and received segments. Thus, TCP-AO emulates a 64-bit sequence number space by combining SNE and the TCP sequence number.

TCP-AO Format

TCP-AO has the following TLV format in the options sequence of a TCP segment:

Kind (1B) = 29	Length (1B)	KeyID (1B)	RNextKeyID (1B)
MAC (12-16B)			
MAC			
MAC			
MAC			

The fields of the TLV format are as follows:

- Kind: Indicates TCP-AO with a value of 29.
- Length: Indicates the length of the TCP-AO sequence.
- KeyID: The send identifier of the Primary Key Tuples that was used to generate the traffic keys.
- RNextKeyID: The receive identifier of the Primary Key Tuples that is ready to be used to authenticate received segments.
- MAC: The MAC computed for the TCP segment data and the prefixed pseudo header.

TCP-AO Key Rollover

TCP-AO keys are valid for a defined duration configured using the send-lifetime and accept-lifetime properties. If send-lifetime and accept-lifetime are not configured for a key, the key has infinite send and accept lifetimes. Key rollover is initiated based on the send lifetimes of keys. As part of key rollover, a key that is valid and has the longest send lifetime into the future is selected as the active key.

When key rollover is initiated, one of the peer routers, say Router A, indicates that the rollover is necessary. To indicate that the rollover is necessary, Router A sets the RNextKeyID to the receive identifier of the new Primary Key Tuples to be used. On receiving the TCP segment, the peer router, say Router B, finds the Primary Key Tuples indicated by the RNextKeyID in the TCP-AO payload. If the key is available and valid, Router B sets the current key to the new Primary Key Tuples. After Router B has rolled over, Router A also sets the current key to the new Primary Key Tuples.

Key rollover can be initiated by one of the following methods:

- Rollover on send-lifetime expiry
- Rollover with overlapping send-lifetimes

If you do not configure a new key that can be activated before the expiry of the current key, the key may time out and expire. Such an expiry can cause retransmissions with the peer router rejecting segments authenticated

with the expired key. The connection may fail due to Retransmission Time Out (RTO). When new valid keys are configured, a new connection is established.

Note

- Key rollover is based only on send lifetimes of keys.
- Key rollover is only supported within a key chain.
- Forced deletion of a key in use does not trigger key rollover.
- From among the keys in a key chain, the key with the longest send lifetime into the future is selected as the active key during a rollover.

Restrictions for TCP Authentication Option

- The send-id and recv-id of each key in the key chain must be unique. Because send-id and recv-id must be chosen from the range 0 to 255, the TCP-AO key chain can have a maximum of 256 keys.
- Only one keychain can be associated with an application connection. Rollover is always performed within the keys in this keychain.
- TCP-AO does not allow the modification of a key in use. Modify a key after disassociating the key from the connection.
- If the key in use expires, expect segment loss until a new key that has a valid lifetime is configured on each side and keys rollover.

How to Configure TCP Authentication Option

Configure TCP Key Chain and Keys

Configure TCP-AO key chain and keys on both the peers communicating through a TCP connection.

Note

- Ensure that the key-string, send-lifetimes, cryptographic-algorithm, and ids of keys match on both peers.
- Ensure that the send-id on a router matches the recv-id on the peer router. We recommend using the same id for both the parameters unless there is a need to use separate key spaces.
- The send-id and recv-id of a key cannot be reused for another key in the same key chain.
- Do not modify properties of a key in use, except when you need to modify the send-lifetime of the key to trigger rollover. Before modifying properties other than send-lifetime, disassociate the key from the TCP connection.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **key chain** *key-chain-name* **tcp**

Example:

```
Device(config)# key chain kc1 tcp
```

Creates a TCP-AO key chain of with a specified name and enters the TCP-AO key chain configuration mode.

The key chain name can have a maximum of 256 characters.

Step 4 **key** *key-id*

Example:

```
Device(config-keychain-tcp)# key 10
```

Creates a key with the specified key-id and enters the TCP-AO key chain key configuration mode.

The key-id must be in the range from 0 to 2147483647.

Note The key-id has only local significance. It is not part of the TCP Authentication Option.

Step 5 **send-id** *send-identifier*

Example:

```
Device(config-keychain-tcp-key)# send-id 218
```

Specifies the send identifier for the key.

The send-identifier must be in the range from 0 to 255.

Step 6 **recv-id** *receiver-identifier*

Example:

```
Device(config-keychain-tcp-key)# recv-id 218
```

Specifies the receive identifier for the key.

The receive-identifier must be in the range from 0 to 255.

Step 7 **cryptographic-algorithm** {**aes-128-cmac** | **hmac-sha-1** | **hmac-sha-256**}

Example:

```
Device(config-keychain-tcp-key)# cryptographic-algorithm hmac-sha-1
```

Specifies the algorithm to be used to compute MACs for TCP segments.

aes-128-cmac	AES-128-CMAC-96: Configures AES-128-CMAC as a cryptographic algorithm with a digest size of 12 bytes.
hmac-sha-1	HMAC-SHA1-96: Configures HMAC-SHA1-96 as a cryptographic algorithm with a digest size of 12 bytes.
hmac-sha-256	HMAC-SHA-256: Configures HMAC-SHA-256 as a cryptographic algorithm with a digest size of 32 bytes.

Step 8 (Optional) **include-tcp-options**

Example:

```
Device(config-keychain-tcp-key)# include-tcp-options
```

This flag indicates whether TCP options other than TCP-AO must be used to calculate MACs.

With the flag enabled, the content of all options, in the order present, is included in the MAC and TCP-AO's MAC field is zero-filled.

When the flag is disabled, all options other than TCP-AO are excluded from MAC calculations.

By default, this flag is disabled.

Step 9 **send-lifetime** [**local**] *start-time* {**infinite** | *end-time* | **duration** *seconds*}

Example:

```
Device(config-keychain-tcp-key)# send-lifetime local 12:00:00 28 Feb 2018 duration 20
```

Specifies the time for which the key is valid to be used for TCP-AO authentication in the send direction.

Use the **local** keyword to specify the start-time in the local time zone. By default, the start-time corresponds to UTC time.

Step 10 **key-string** *master-key*

Example:

```
Device(config-keychain-tcp-key)# key-string abcde
```

Specifies the primary-key for deriving traffic keys.

The primary-keys must be identical on both the peers. If the primary-keys do not match, authentication fails and segments may be rejected by the receiver.

Step 11 (Optional) **accept-ao-mismatch**

Example:

```
Device(config-keychain-tcp-key)# accept-ao-mismatch
```

This flag indicates whether the receiver should accept segments for which the MAC in the incoming TCP AO does not match the MAC generated on the receiver.

Note Use this configuration with caution. This configuration disables TCP-AO functionality and key rollover on associated connections.

Step 12 **end**

Example:

```
Device(config-keychain-tcp-key)# end
```

Exits TCP-AO key chain key configuration mode and returns to privileged EXEC mode.

Verifying TCP-AO Key Chain and Key Configuration

Use the **show key chain** *key-chain-name* command in the privileged EXEC mode to display information about a TCP-AO key chain and keys, and association with TCBs.

```
Router# show key chain key-chain-name
```

```
Router1# show key chain kc1
Key-chain kc1:
    TCP key chain
    key 7893 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (12:32:00 IST Nov 9 2018) - (10:30:00 IST Dec 30 2019) [valid now]
        send lifetime (13:05:00 IST Jan 12 2019) - (10:31:00 IST Dec 30 2019) [valid now]
        send-id - 218
        recv-id - 218
        include-tcp-options
        MKT ready - true
        MKT preferred - true
        MKT in-use - true
        MKT id - 7893
        MKT send-id - 218
        MKT recv-id - 218
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - true
        MKT accept AO mismatch - false
            TCB - 0x7FBD68361838
            curr key - 7893
            next key - 7893
```

Verifying TCP-AO Key Chain Information in the TCB

Use the **show tcp tcb** *address-of-tcb* command in the privileged EXEC mode to display information about TCP-AO in the Transmission Control Block. Obtain *address-of-tcb*(the hexadecimal address of the TCB) from the output of the **show key chain** *key-chain-name* command.

```
Router# show tcp tcb address-of-tcb
```

```
Router1# show tcp tcb 7FBD68361838
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 1.0.2.1, Local port: 40125
Foreign host: 1.0.2.2, Foreign port: 5555
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x2818B07):
Timer          Starts    Wakeups            Next
Retrans             1          0             0x0
TimeWait            0          0             0x0
```

```
AckHold             1          0             0x0
SendWnd             0          0             0x0
KeepAlive        6651          0       0x281AC36
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
ProcessQ            0          0             0x0

iss: 3307331702  snduna: 3307331703  sndnxt: 3307331703
irs:  725047078  rcvnxt:  725047079

sndwnd:   4128  scale:      0  maxrcvwnd:   4128
rcvwnd:   4128  scale:      0  delrcvwnd:      0

SRTT: 125 ms, RTTO: 2625 ms, RTV: 2500 ms, KRTT: 0 ms
minRTT: 15 ms, maxRTT: 1000 ms, ACK hold: 200 ms
uptime: 40996359 ms, Sent idletime: 6505 ms, Receive idletime: 6505 ms
Status Flags: active open
Option Flags: keepalive running, nagle, Retrans timeout
IP Precedence value : 0

TCP AO Key chain: kc1

TCP AO Current Key:
 Id: 7893, Send-Id: 218, Recv-Id: 218
 Include TCP Options: Yes*
 Accept AO Mismatch: No*

TCP AO Next Key:
 Id: 7893, Send-Id: 218, Recv-Id: 218
 Include TCP Options: Yes*
 Accept AO Mismatch: No*

Datagrams (max data segment is 1460 bytes):
Rcvd: 4372 (out of order: 0), with data: 0, total data bytes: 0
Sent: 4372 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with
data: 0, total data bytes: 0

 Packets received in fast path: 0, fast processed: 0, slow path: 0
 fast lock acquisition failures: 0, slow path: 0
TCP Semaphore      0x7FBD6801B2E0  FREE

* - Derived from Key
```

Configuring Key Rollover on Send Lifetime Expiry

Configure a new key in the key chain such that the key becomes active on the expiry of the send-lifetime of the currently active key. The examples in the following steps show sample configurations on two peer routers, Router 1 and Router 2. In these examples, the active key has an id of 7890 and the new key has an id of 7891.

Step 1 Identify the active key on both peer routers.

Example:

Identify active key on Router 1:

```
Router1#show run | sec key
key chain kc1 tcp
 key 7890
```

```
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
```

Identify active key on Router 2:

```
Router2# show run | sec key
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
```

Step 2 Configure the new key on both peer routers.

Example:

Configure new key on Router 1:

```
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
 key 7891
  send-id 216
  recv-id 216
  cryptographic-algorithm hmac-sha-1
  key-string fghij
```

Configure new key on Router 2:

```
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
 key 7891
  send-id 216
  recv-id 216
  cryptographic-algorithm hmac-sha-1
  key-string fghij
```

When the send-lifetime of the active key expires, the new key is activated. Syslog messages are displayed indicating rollover to the new key.

Step 3 Reduce the send-lifetimes of active keys on the peer routers.

Example:

Reduce send-lifetime of the active key on Router 1:

```
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
   send-lifetime local 10:00:00 Jun 24 2019 13:45:00 Jun 24 2019
 key 7891
  send-id 216
  recv-id 216
```

```
  cryptographic-algorithm hmac-sha-1
  key-string fghij
```

Reduce send-lifetime of active key on Router 2:

```
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
   send-lifetime local 10:00:00 Jun 24 2019 13:45:00 Jun 24 2019
 key 7891
  send-id 216
  recv-id 216
  cryptographic-algorithm hmac-sha-1
  key-string fghij
```

Step 4 Verify the send-lifetimes of the currently active and new keys on the peer routers.

Example:

Verify send-lifetimes of the keys on Router 1:

```
Router1# sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7891
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (10:00:00 IST Jun 24 2019) - (13:45:00 IST Jun 24 2019) --- [valid now]
        send-id - 215
        recv-id - 215
        MKT ready - true
        MKT preferred - false
        MKT in-use - true
        MKT id - 7890
        MKT send-id - 215
        MKT recv-id - 215
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
            TCB - 0x7FC0EC097AC0
            curr key - 7890
            next key - 7890
            TCB - 0x7FC0EBBE7600
            curr key - 7890
            next key - 7890
    key 7891 -- text ”fghij"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) --- [valid now]
        send-id - 216
        recv-id - 216
        MKT ready - true
        MKT preferred - true
        MKT in-use - false
        MKT id - 7891
        MKT send-id - 216
        MKT recv-id - 216
        MKT alive (send) - true
        MKT alive (recv) - true
```

```
        MKT include TCP options - false
        MKT accept AO mismatch - false
```

Verify send-lifetimes of the keys on Router 2:

```
Router2# sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7891
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (10:00:00 IST Jun 24 2019) - (13:45:00 IST Jun 24 2019) --- [valid now]
        send-id - 215
        recv-id - 215
        MKT ready - true
        MKT preferred - false
        MKT in-use - true
        MKT id - 7890
        MKT send-id - 215
        MKT recv-id - 215
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
            TCB - 0x7FB6BEF4CC10
            curr key - 7890
            next key - 7890
            TCB - 0x7FB6BEAA7B28
            curr key - 7890
            next key - 7890
    key 7891 -- text ”fghij"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) --- [valid now]
        send-id - 216
        recv-id - 216
        MKT ready - true
        MKT preferred - true
        MKT in-use - false
        MKT id - 7891
        MKT send-id - 216
        MKT recv-id - 216
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
```

Step 5 Verify key rollover on the routers using the **show key chain** command.

Example:

Verify key rollover on Router 1:

```
Router1#
*Jun 24 08:15:00.000: %TCP-6-AOKEYSENDEXPIRED: TCP AO Keychain kc1 key 7890 send lifetime expired
*Jun 24 08:15:00.000: %TCP-6-AOROLLOVER: TCP AO Keychain kc1 rollover from key 7890 to key 7891

Router1#sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7891
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
```

```
        send lifetime (10:00:00 IST Jun 24 2019) - (13:45:00 IST Jun 24 2019)
        send-id - 215
        recv-id - 215
        MKT ready - true
        MKT preferred - false
        MKT in-use - false
        MKT id - 7890
        MKT send-id - 215
        MKT recv-id - 215
        MKT alive (send) - false
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
    key 7891 -- text ”fghij"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
        send-id - 216
        recv-id - 216
        MKT ready - true
        MKT preferred - true
        MKT in-use - true
        MKT id - 7891
        MKT send-id - 216
        MKT recv-id - 216
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
            TCB - 0x7FC0EBBE7600
            curr key - 7891
            next key - 7891
            TCB - 0x7FC0EC097AC0
            curr key - 7891
            next key - 7891
```

Verify key rollover on Router 2:

```
Router2#
*Jun 24 08:15:00.000: %TCP-6-AOKEYSENDEXPIRED: TCP AO Keychain kc1 key 7890 send lifetime expired
*Jun 24 08:15:00.000: %TCP-6-AOROLLOVER: TCP AO Keychain kc1 rollover from key 7890 to key 7891

Router2#sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7891
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (10:00:00 IST Jun 24 2019) - (13:45:00 IST Jun 24 2019)
        send-id - 215
        recv-id - 215
        MKT ready - true
        MKT preferred - false
        MKT in-use - false
        MKT id - 7890
        MKT send-id - 215
        MKT recv-id - 215
        MKT alive (send) - false
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
    key 7891 -- text ”fghij"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
```

```
send lifetime (always valid) - (always valid) [valid now]
send-id - 216
recv-id - 216
MKT ready - true
MKT preferred - true
MKT in-use - true
MKT id - 7891
MKT send-id - 216
MKT recv-id - 216
MKT alive (send) - true
MKT alive (recv) - true
MKT include TCP options - false
MKT accept AO mismatch - false
    TCB - 0x7FB6BEAA7B28
    curr key - 7891
    next key - 7891
    TCB - 0x7FB6BEF4CC10
    curr key - 7891
    next key - 7891
```

Configuring Key Rollover with Overlapping Send Lifetimes

Configure a new key in the key chain such that the currently active key and new key have overlapping send-lifetime values. Also, configure the send-lifetime of the new key such that it extends longer into the future than the send-lifetime of the currently active key. During key rollover, the key with the longest send-lifetime into the future is selected as the active key. Thus, when the send-lifetime of the new key begins, the key becomes active.

The examples in the following steps show sample configurations on two peer routers, Router 1 and Router 2. In these examples, the active key has an id of 7890 and the new key has an id of 7891.

Step 1 Identify the active key on both peer routers.

Example:

Identify active key on Router 1:

```
Router1# show run | sec key
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
  send-lifetime local 10:00:00 Jun 24 2019
   10:00:00 Aug 24 2019
```

Identify active key on Router 2:

```
Router2# show run | sec key
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
  send-lifetime local 10:00:00 Jun 24 2019
   10:00:00 Aug 24 2019
```

Step 2 Configure a new key with an overlapping send-lifetime on both peer routers.

Example:

Configure new key on Router 1:

```
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
  send-lifetime local 10:00:00 Jun 24 2019 10:00:00 Aug 24 2019
 key 7891
  send-id 216
  recv-id 216
  cryptographic-algorithm hmac-sha-1
  key-string fghij
  send-lifetime local 21:50:00 Jun 24 2019 11:00:00 Aug 24 2019
```

Configure new key on Router 2:

```
key chain kc1 tcp
 key 7890
  send-id 215
  recv-id 215
  cryptographic-algorithm hmac-sha-1
  key-string abcde
  send-lifetime local 10:00:00 Jun 24 2019 10:00:00 Aug 24 2019
 key 7891
  send-id 216
  recv-id 216
  cryptographic-algorithm hmac-sha-1
  key-string fghij
  send-lifetime local 21:50:00 Jun 24 2019 11:00:00 Aug 24 2019
```

When the send-lifetime of the new key starts, the new key is activated. Syslog messages are displayed indicating rollover to the new key.

Step 3 Verify that the send-lifetimes of the currently active and new keys are overlapping.

Example:

Verify send-lifetimes of the keys on Router 1:

```
Router1# sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7890
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (10:00:00 IST Jun 24 2019) - (10:00:00 IST Aug 24 2019)--- [valid now]
        send-id - 215
        recv-id - 215
        MKT ready - true
        MKT preferred - true
        MKT in-use - true
        MKT id - 7890
        MKT send-id - 215
        MKT recv-id - 215
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
```

```
            TCB - 0x7F8352155318
            curr key - 7890
            next key - 7890
            TCB - 0x7F8352FF37F0
            curr key - 7890
            next key - 7890
    key 7891 -- text "fghij"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (21:50:00 IST Jun 24 2019) - (11:00:00 IST Aug 24 2019)
        send-id - 216
        recv-id - 216
        MKT ready - true
        MKT preferred - false
        MKT in-use - false
        MKT id - 7891
        MKT send-id - 216
        MKT recv-id - 216
        MKT alive (send) - false
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
```

Verify send-lifetimes of the keys on Router 2:

```
Router2#sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7890
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (10:00:00 IST Jun 24 2019) - (10:00:00 IST Aug 24 2019)--- [valid now]
        send-id - 215
        recv-id - 215
        MKT ready - true
        MKT preferred - true
        MKT in-use - true
        MKT id - 7890
        MKT send-id - 215
        MKT recv-id - 215
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
            TCB - 0x7F5FCD185150
            curr key - 7890
            next key - 7890
            TCB - 0x7F5FD2734C48
            curr key - 7890
            next key - 7890
    key 7891 -- text "fghij"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (21:50:00 IST Jun 24 2019) - (11:00:00 IST Aug 24 2019)
        send-id - 216
        recv-id - 216
        MKT ready - true
        MKT preferred - false
        MKT in-use - false
        MKT id - 7891
        MKT send-id - 216
        MKT recv-id - 216
        MKT alive (send) - false
        MKT alive (recv) - true
```

```
        MKT include TCP options - false
        MKT accept AO mismatch – false
```

Step 4 Verify key rollover on the routers using the **show key chain** command.

Example:

Verify key rollover on Router 1:

```
Router1#
*Jun 24 16:20:00.000: %TCP-6-AOROLLOVER: TCP AO Keychain kc1 rollover from key 7890 to key 7891
Router1#sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7891
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (10:00:00 IST Jun 24 2019) - (10:00:00 IST Aug 24 2019) [valid now]
        send-id - 215
        recv-id - 215
        MKT ready - true
        MKT preferred - false
        MKT in-use - false
        MKT id - 7890
        MKT send-id - 215
        MKT recv-id - 215
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
    key 7891 -- text "fghij"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (21:50:00 IST Jun 24 2019) - (11:00:00 IST Aug 24 2019) [valid now]
        send-id - 216
        recv-id - 216
        MKT ready - true
        MKT preferred - true
        MKT in-use - true
        MKT id - 7891
        MKT send-id - 216
        MKT recv-id - 216
        MKT alive (send) - true
        MKT alive (recv) - true
        MKT include TCP options - false
        MKT accept AO mismatch - false
            TCB - 0x7F8352FF37F0
            curr key - 7891
            next key - 7891
            TCB - 0x7F8352155318
            curr key - 7891
            next key - 7891
```

Verify key rollover on Router 2:

```
Router2#
*Jun 24 16:20:00.000: %TCP-6-AOROLLOVER: TCP AO Keychain kc1 rollover from key 7890 to key 7891
Router2#sh key chain
Key-chain kc1:
    TCP key chain
    Preferred MKT id - 7891
    key 7890 -- text "abcde"
        cryptographic-algorithm: hmac-sha-1
        accept lifetime (always valid) - (always valid) [valid now]
```

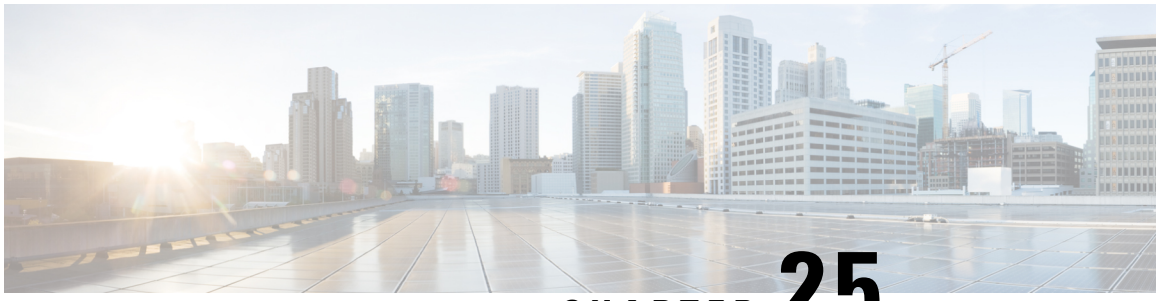
```
    send lifetime (10:00:00 IST Jun 24 2019) - (10:00:00 IST Aug 24 2019) [valid now]
    send-id - 215
    recv-id - 215
    MKT ready - true
    MKT preferred - false
    MKT in-use - false
    MKT id - 7890
    MKT send-id - 215
    MKT recv-id - 215
    MKT alive (send) - true
    MKT alive (recv) - true
    MKT include TCP options - false
    MKT accept AO mismatch - false
key 7891 -- text "fghij"
    cryptographic-algorithm: hmac-sha-1
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (21:50:00 IST Jun 24 2019) - (11:00:00 IST Aug 24 2019) [valid now]
    send-id - 216
    recv-id - 216
    MKT ready - true
    MKT preferred - true
    MKT in-use - true
    MKT id - 7891
    MKT send-id - 216
    MKT recv-id - 216
    MKT alive (send) - true
    MKT alive (recv) - true
    MKT include TCP options - false
    MKT accept AO mismatch - false
        TCB - 0x7F5FD2734C48
        curr key - 7891
        next key - 7891
        TCB - 0x7F5FCD185150
        curr key - 7891
        next key - 7891
```

Feature Information for TCP Authentication Option

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 27: Feature Information for TCP Authentication Option



CHAPTER 25

Configuring On-Demand Routing

The On-Demand Routing feature provides IP routing for stub sites, with minimum cost. The cost of a general, dynamic routing protocol is avoided without incurring the configuration and management burden of static routing.

Prerequisites for Configuring On-Demand Routing

Cisco Discovery Protocol must be enabled.

Restrictions for Configuring On-Demand Routing

No IP routing protocol can be configured on the stub router.

Information About On-Demand Routing

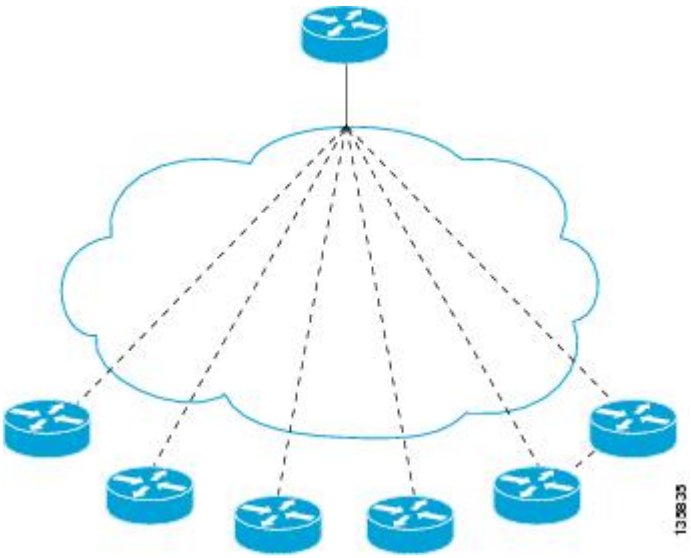
Benefits of On-Demand Routing

This module describes how to configure On-Demand Routing (ODR). The ODR feature provides IP routing for stub sites, with minimum cost. The cost of a general, dynamic routing protocol is avoided without incurring the configuration and management burden of static routing.

Stub Networks

A stub router can be considered a spoke router in a hub-and-spoke network topology--as shown in the figure below--where the only router to which the spoke is adjacent is the hub router. In such a network topology, the IP routing information required to represent this topology is fairly simple. These stub routers commonly have a WAN connection to the hub router, and a small number of LAN network segments (stub networks) are directly connected to the stub router. These stub networks might consist only of end systems and the stub router, and therefore do not require the stub router to learn any dynamic IP routing information.

Figure 15: Hub-And-Spoke Network Topology Example



Overview of On-Demand Routing

ODR allows you to easily install IP stub networks where the hubs dynamically maintain routes to the stub networks. This installation is accomplished without requiring the configuration of an IP routing protocol on the stubs. In fact, from the standpoint of ODR, a router is automatically considered to be a stub when no IP routing protocols have been configured.

A stub router that supports the ODR feature advertises IP prefixes corresponding to the IP networks configured on all directly connected interfaces. If the interface has multiple logical IP networks configured, only the primary IP network is advertised through ODR. Because ODR advertises IP prefixes and not simply IP network numbers, ODR is able to carry variable-length subnet mask (VLSM) information.

Once ODR is enabled on a hub router, the hub router begins installing stub network routes in the IP forwarding table. The hub router also can be configured to redistribute these routes into any configured dynamic IP routing protocols.

ODR uses the Cisco Discovery Protocol to carry minimal routing information between the hub and stub routers. The stub routers send IP prefixes to the hub router. The hub router provides default route information to the stub routers, thereby eliminating the need to configure a default route on each stub router.

How to Configure On-Demand Routing

Enabling ODR

Once ODR is enabled on a hub router, the hub router begins installing stub network routes in the IP forwarding table. The hub router also can be configured to redistribute these routes into any configured dynamic IP routing protocols.

To enable ODR on a hub router, perform the steps in this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router odr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router odr** **Example:** `Router(config)# router odr`	Enables ODR on a Cisco router, and places the router in router configuration mode.

Disabling the Propagation of ODR Stub Routing Information

ODR uses Cisco Discovery Protocol to carry minimal routing information between the hub and stub routers, allowing stub routers to send IP prefixes to the hub router. Perform the steps in this task to disable the propagation of ODR stub routing information by disabling CDP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no cdp run**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no cdp run** **Example:** `Router(config)# no cdp run`	Disables Cisco Discovery Protocol.

Disabling the Propagation of ODR Stub Routing Information on a Specified Interface

On stub routers that support the ODR feature, the stub router advertises IP prefixes corresponding to the IP networks configured on all directly connected interfaces. Perform the steps in this task to disable the propagation of ODR stub routing information on a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no cdp enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**no cdp enable** **Example:** `Router(config-if)# no cdp enable`	Disables Cisco Discovery Protocol on an interface.

Filtering ODR Information

The hub router will attempt to populate the IP routing table with ODR routes as they are learned dynamically from stub routers. The IP next hop for these routes is the IP address of the neighboring router as advertised through Cisco Discovery Protocol. Use IP filtering to limit the network prefixes that the hub router will permit to be learned dynamically through ODR.

In this example, the ACL filters the following Class A network prefixes:

```
access-list 101 permit 10.48.0.3
access-list 101 deny 10.48.0.0  0.0.255.255
access-list 101 permit 10.0.0.0  0.255.255.255
interface gigabitethernet 0/0/0
 ip access-group 2 in
```

To filter ODR information, perform the steps in this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
4. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
5. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
6. router odr
7. **distribute-list** [[*access-list-number* | *name*] | [**route-map** *map-tag*]] **in** [*interface-type* | *interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Router(config)# access-list 101 permit 10.48.0.3`	Access-list 101 permits the IP address 10.48.0.3.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Router(config)# access-list 101 deny 10.48.0.0 0.0.255.255`	Access-list 101 denies the IP address 10.48.0.0 0.0.255.255.
Step 5	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Router(config)# access-list 101 permit 10.0.0.0 0.255.255.255`	Access-list 101 permits the IP address 10.0.0.0 0.255.255.255.
Step 6	router odr **Example:** `Router(config)# router odr`	Enables ODR and enters router configuration mode.
Step 7	**distribute-list** [[*access-list-number* | *name*] | [**route-map** *map-tag*]] **in** [*interface-type* | *interface-number*] **Example:** `Router(config-router)# distribute-list 101 in`	Filters ODR information on the hub router.

Redistributing ODR Information into the Dynamic Routing Protocol of the Hub

The exact command syntax needed to redistribute ODR information into the dynamic routing protocol of the hub depends upon the routing protocol into which ODR is being redistributed. See the "Redistributing Routing Information" section in the "Configuring IP Routing Protocol-Independent Features" module for further information.

Reconfiguring Cisco Discovery Protocol or ODR Timers

By default, Cisco Discovery Protocol sends updates every 60 seconds. This update interval may not be frequent enough to provide fast reconvergence of IP routes on the hub router side of the network. A faster reconvergence rate may be necessary if the stub connects to one of several hub routers via asynchronous interfaces such as modem lines.

ODR expects to receive periodic Cisco Discovery Protocol updates containing IP prefix information. When ODR fails to receive such updates for routes that it has installed in the routing table, these ODR routes are first marked invalid and eventually removed from the routing table. (By default, ODR routes are marked invalid after 180 seconds and are removed from the routing table after 240 seconds.) These defaults are based on the default Cisco Discovery Protocol update interval. Configuration changes made to either the Cisco Discovery Protocol or ODR timers should be reflected through changes made to both.

To reconfigure Cisco Discovery Protocol or ODR timers, perform the steps in this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cdp timer** *seconds*
4. router odr
5. **timers basic** *update invalid holddown flush sleeptime*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cdp timer** *seconds* **Example:** `Router(config)# cdp timer 80`	Specifies how often the Cisco IOS XE software sends Cisco Discovery Protocol updates.
Step 4	router odr **Example:** `Router(config)# router odr`	Enables ODR and enters router configuration mode.
Step 5	**timers basic** *update invalid holddown flush sleeptime* **Example:** `Router(config-router)# timers basic 5 15 15 30`	Adjusts ODR network timers.

Using Dialer Map Statements to Direct Cisco Discovery Protocol Broadcast Packets

For interfaces that specify dialer mappings, Cisco Discovery Protocol packets will make use of dialer map configuration statements that pertain to the IP protocol. Because Cisco Discovery Protocol packets are always broadcast packets, these dialer map statements must handle broadcast packets, typically through use of the **dialer map** command with the **broadcast** keyword. The **dialer string** command in interface configuration mode may also be used.

On dial-on-demand (DDR) routing interfaces, certain kinds of packets can be classified as interesting. These interesting packets can cause a DDR connection to be made or cause the idle timer of a DDR interface to be reset. For the purposes of DDR classification, Cisco Discovery Protocol packets are considered uninteresting. This classification occurs even while Cisco Discovery Protocol is making use of dialer map statements for IP, where IP packets are classified as interesting.

The following task describes how to use dialer map statements to direct Cisco Discovery Protocol broadcast packets.

or

dialer string *dial-string* [**:** *isdn-subaddress*]

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **dialer map** *protocol-keyword protocol-next-hop-address* [**broadcast** | **class** *dialer-map-class-name* | **modem-script** *modem-regular-expression* | **vrf** *vrf-name* | **name** *host-name* | **spc** | **speed 56** | **speed 64** | **system-script** *system-regular-expression* | *dial-string*[**:** *isdn-subaddress*]]
   -
   -
   - **dialer string** *dial-string* [**:** *isdn-subaddress*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface async 1/0/0`	Configures an interface type, and enters interface configuration mode.
Step 4	Do one of the following: • **dialer map** *protocol-keyword protocol-next-hop-address* [**broadcast** | **class** *dialer-map-class-name* | **modem-script** *modem-regular-expression* | **vrf** *vrf-name* | **name** *host-name* | **spc** | **speed 56** | **speed 64** | **system-script** *system-regular-expression* | *dial-string*[**:** *isdn-subaddress*]] • • • **dialer string** *dial-string* [**:** *isdn-subaddress*] **Example:** `Router(config)# dialer map ip 172.19.2.5 speed 56`	Configures an asynchronous interface to call multiple sites or to receive calls from multiple sites. Specifies the string (telephone number) to be called for interfaces calling a single site.

Configuration Examples for On-Demand Routing

Enabling ODR and Filtering ODR Information Example

The following example shows how to enable ODR on a Cisco router and enable filtering of ODR information. The configuration example for filtering ODR information causes the hub router to accept only advertisements for IP prefixes about (or subnets of) the Class C network 192.168.1.0:

```
Router(config)# access-list 101 permit ip host 10.0.0.1 192.168.1.0 0.0.0.255

Router(config)# access-list 101 permit ip 10.0.10.2 255.0.0.0 192.168.2.0 0.0.0.255
Router(config)# router odr
Router(config-router)# distribute-list 101 in
Router(config-router)# end
```

Disabling ODR on a Specified Interface Example

The following example shows how to disable ODR on an interface:

```
Router# configure terminal
Router(config)# interface GigabitEthernet 0/0/0

Router(config-if)# no cdp enable
```

Additional References

Related Documents

Related Topic	Document Title
Cisco Discovery Protocol features	"Using Cisco Discovery Protocol" chapter of the *Cisco IOS XE Network Management Configuration Guide*, Release 2
ODR commands	"On-Demand Routing Commands" chapter of the *Cisco IOS IP Routing: ODR Command Reference.*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

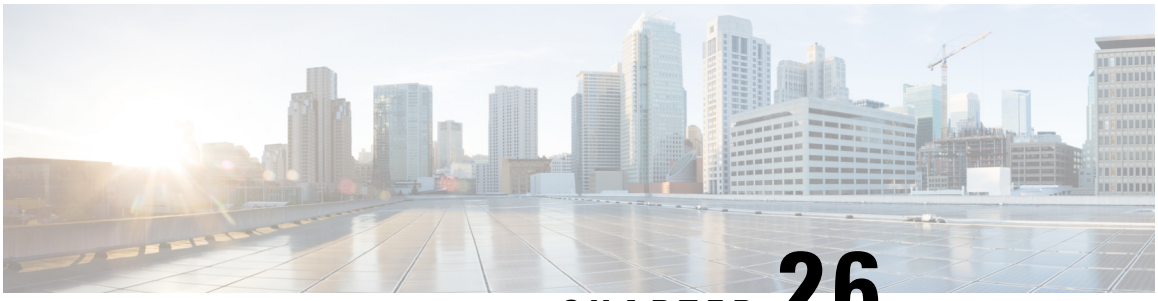
Feature Information for Configuring On-Demand Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 28: Feature Information for Configuring On-Demand Routing

Feature Name	Releases	Feature Information
On-Demand Routing	10.0 12.2(1) 12.2(2)T 15.3(1)S	The On-Demand Routing (ODR) feature provides IP routing for stub sites, with minimum overhead.

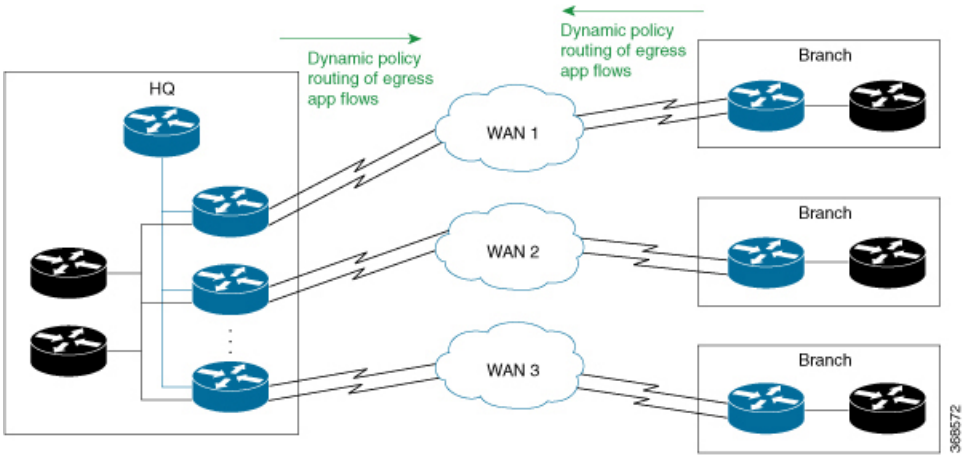


CHAPTER 26

DAPR Overview

Dynamic Application Policy Routing (DAPR) is a WAN-edge egress traffic engineering solution for multi-homed sites. DAPR monitors a WAN link bandwidth and utilization. Also, monitors egress application flow rates in real time and dynamically steers application flows to meet the policy criteria of link preference and link load balancing. DAPR does not have an overlay dependency and therefore cannot manage an overlay or underlay traffic. Typical use cases for DAPR are the WAN edge and the Internet edge.

Figure 16: Dynamic Application Policy Routing



Information about DAPR

This section includes the following topics:

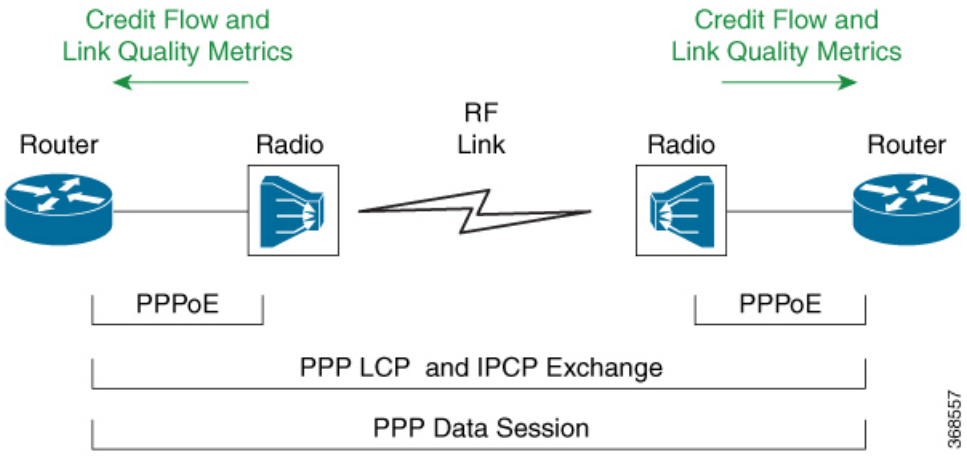
DAPR Fundamentals

1. DAPR is site-local, single-sided, and egress-only:
   - Site-local: DAPR runs independently at each site (Branch, Campus, or Datacenter) with significance only at the local site. DAPR instances running at different sites of an enterprise are completely independent of one another.
   - Single-sided: DAPR has all its functionality and components that are localized at a site. DAPR does not require any components at or any co-ordination with remote sites.
   - Egress-only: DAPR manages only the traffic egressing a site (LAN to WAN). DAPR does not manage ingress traffic (WAN to LAN). More specifically, DAPR only manages the egress flows traversing DAPR-enabled LAN and WAN links.
2. DAPR is for multi-homed sites:
   - DAPR is for sites with multiple WAN links terminating on one or more WAN edge routers that are referred to as DAPR Border-Routers (BR).
   - DAPR provides policy routing of application flows across all the DAPR-enabled WAN links at a site.
3. Role of routing protocols in DAPR:
   - DAPR relies on the routing table (RIB) to determine an application flow destination reachability and hence is independent of routing protocols.
   - The routing protocols' role in DAPR is to make available all possible paths to a destination and not the best path selection. Tune the routing protocol metrics to ensure all possible paths to a destination (not just the best path) are available in the routing table either as equal cost or unequal cost routes.
   - DAPR performs the best path selection for application flows and enforcement.
4. DAPR application flow routing:
   - DAPR dynamic best path selection for application flow-groups is based on:
     - Policy criteria of the link preference and link load balancing:
     - Varying WAN link bandwidth or utilization
     - Varying application flow rates
   - DAPR currently does not monitor the link delay, jitter, and throughput as DAPR does not use any probes.
5. DAPR policy criteria:
   - Link load balancing - Ensures uniform utilization of DAPR. Enables WAN links at a site by dynamically steering application flows across WAN links based on changing link bandwidth or utilization and flow rates.
   - Link preference: Ensure application performance by dynamically steering application flows to specified preferred links.

6. DAPR flow-groups:
   - DAPR identifies application flow-groups based on a 3-tuple of source IP-address, destination IP-address, and DSCP only.
   - DAPR currently does not support the identification of an application flow-groups using NBAR or 5-tuple of source-prefix, destination-prefix, protocol, source-ports, and destination-ports.

7. DAPR supports Radio aware routing (RAR) WAN links:
   - RAR is a solution for the variable bandwidth radio links used in mobile ad hoc networks (MANET). RAR helps in quick detection of neighbors and peers. It also tracks the bandwidth changes of radio links and makes it available to applications such as routing protocols and QoS shapers that rely on a link bandwidth. RAR implementation in Cisco IOS XE Gibraltar 16.11.1 is based on RFC-5578 (PPP over Ethernet (PPPoE) Extensions for Credit Flow and Link Metrics). RAR uses a point-to-point virtual-access interface per peer and updates the virtual-access interface bandwidth value when the corresponding radio link's bandwidth changes.

     Figure 17: Radio Aware Routing



   - DAPR supports RAR and PPPoE virtual access interfaces as DAPR egress interfaces (DAPR-enabled WAN links). DAPR supports RAR bypass mode only.

DAPR Terminology

The following are the terminologies that are used in the DAPR solution:

- Dynamic Application Policy Routing (DAPR): DAPR is the per-site dynamic policy routing solution for the application flows egressing WAN links.
- Route-Manager (RM): DAPR control plane entity at a site that dynamically computes policy conformant routes for the application flows egressing WAN links.
- Border-Router (BR): WAN edge routers at a site that export monitoring information to and enforce the application flow routes computed by the RM.

- Flow-groups: A group of application flows managed by DAPR as a unit. DAPR route computation and enforcement are on a per flow-group basis. Currently, flows are grouped only based on a 3 tuple of source-address, destination-address, and DSCP.
- Link-groups: An arbitrary group of links that specifies the preferred links in a link preference policy.
- DAPR egress interface: A DAPR enabled WAN interface.
- DAPR ingress interface: A DAPR enabled LAN interface. DAPR manages only the flows traversing DAPR ingress and egress interfaces.
- Ingress-BR: BR that receives a flow-group from LAN. Note that Ingress-BR is per flow-group. A flow-group can have one or more Ingress BRs wherein individual flows of a flow-group enter different BRs from the LAN side.
- Egress-BR: BR through which a flow-group leaves the site through WAN links. Note that Egress-BR is per flow-group. A flow-group can have a single Egress-BR even if the Ingress-BRs are many.
- Locally forwarded flow-groups: Flow-groups for which Ingress-BR and the computed Egress-BR is the same.
- Inter-BR forwarded flows: Flow-groups for which Ingress-BR and the computed Egress-BR are not the same. Such flows are forwarded from Ingress-BR to Egress-BR over the inter-BR IP or GRE tunnel that is referred to as auto-tunnel.
- Auto-tunnel: IP/GRE tunnel between each pair of BRs that are automatically created by DAP.
- Link out-of-policy (OOP) - A condition when DAPR egress exceeds the maximum percentage utilization threshold that is specified in the DAPR policy on RM.
    - Link soft-OOP: OOP link but not exceeding link capacity
    - Link hard-OOP: OOP link exceeding link capacity

DAPR Topologies

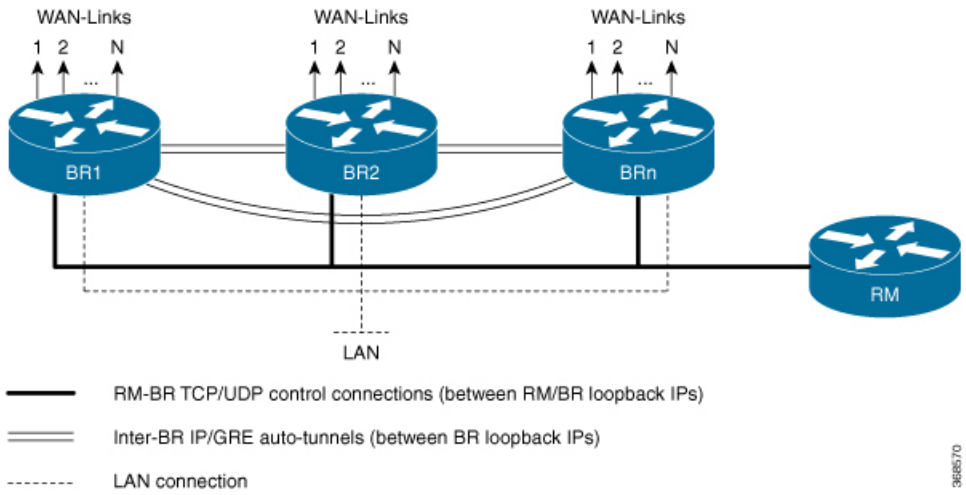
DAPR supports two topologies at a site:

- Standalone RM and BRs
- Co-located RM and BR

Standalone Route Manager and Border Routers

In this topology, Route-Manager (RM) and Border-Routers (BR) are deployed on separate routers. This is commonly used at large sites such as Campus or Headquarters, Datacenter, or large branch sites.

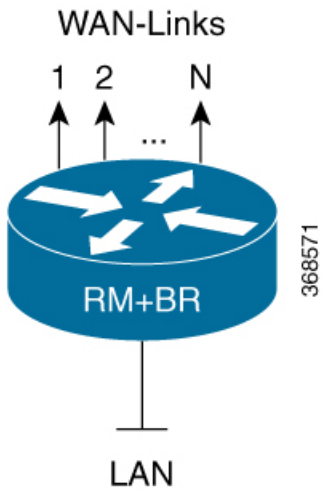
Figure 18: DAPR Standalone RM and BR



Co-located Route Manager and Border Routers

In this topology, RM and BR are deployed on a single router. This is commonly used at small sites with a single WAN edge router such as small branch sites.

Figure 19: DAPR Co-located RM and BR



DAPR Components

DAPR solution comprises the following control and data plane functions:

DAPR Control Plane

1. Collection of site-wide metrics for the flow-route computation.
    - Flows and flow-metrics (byte or packet count and input or output interfaces)
    - Flow destination reachability information

- WAN link metrics (such as bandwidth & utilization)

2. Computation of per flow-group policy routes based on the site-wide metrics.
3. Synchronized programming of the per flow-group policy-route decisions (forwarding state) on the WAN edge routers (BRs).

DAPR Data Plane

1. Enforcement of the per flow-group policy-routes bypassing normal routing.
2. Inter-BR traffic forwarding to enforce policy-route decisions where the Ingress and Egress BRs for a traffic flow group are not the same.

DAPR comprises of the following entities and inter-communication:

Route Manager

Route-manager is a control plane entity that performs following functions:

1. Registration of BRs:
   a. Authentication and authorization of BRs
   b. Push policy parameters (e.g. link thresholds) and neighbor-BR information
2. Periodic processing.
   a. Information pull from BRs:
      - Bandwidth and utilization of DAPR egress interfaces.
      - Routes for prefixes reachable through DAPR egress interfaces.
      - Egress flows on DAPR egress interfaces and flow parameters.
   b. Route computation:
      - Best route computation for new application flow groups.
      - Route re-computation for existing out-of-policy flow groups.
      - Route re-computation for existing flow groups that are impacted by events such as WAN link down, route delete and so on.
   c. Route push to BRs for enforcement:
      - Flow-group routes are pushed only to ingress-BRs (BRs receiving the flow-group from LAN).
      - Flow-group routes specify egress BR and interface through which the flows must egress. Flow-groups that must egress through other BRs are forwarded over inter-BR auto-tunnels.
3. Event processing:
   a. Processing of RM and BR events.
   b. Route re-computation for relocation of flow groups.

c. Push re-computed routes to BRs for enforcement.

Border Router

Border router performs the following:

1. Registration with RM:
   a. Register DAPR egress and ingress interfaces (DAPR-enabled WAN and LAN interfaces).
   b. Create auto-tunnels to neighbor BRs learnt from RM, for inter-BR traffic forwarding.
2. Provide monitoring information to RM (periodically pulled by RM):
   a. Bandwidth and utilization of DAPR egress interfaces.
   b. Prefixes reachable through DAPR egress interfaces.
   c. Application flow groups egressing DAPR egress interfaces.
      • State of auto-tunnels to neighbor BRs.
3. Event notifications to RM:
   a. Reachability events such as DAPR egress down and prefix unreachable.
   b. Threshold violation events.
   c. Inter-BR reachability such as auto-tunnel down.
4. Enforcement of application flow-group routes received from RM.
   a. Enforce routes by bypassing routing and using pre-routing.
   b. For routes with non-local egresses, forward traffic to egress/neighbor BRs over auto-tunnels.

Route Manager and Border Router Communication

DAPR control connections are between the RM and BR loopback IP addresses. DAPR uses two protocols for RM and BR control communication.

- TCP based control protocol is used for registration, information pull and route push by RM and event notifications from BRs.
- UDP based FNF (Flexible Netflow v9) protocol is used by BRs to periodically export the egress flows on DAPR egress interfaces.

Figure 20: DAPR Registration

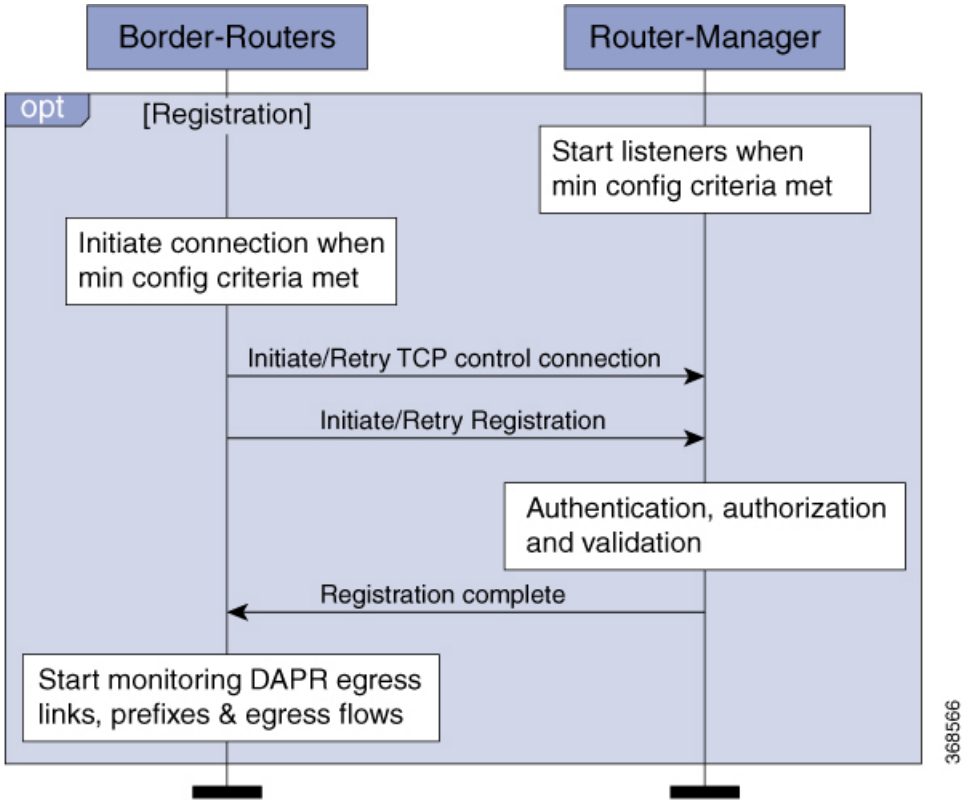


Figure 21: DAPR Periodic Processing

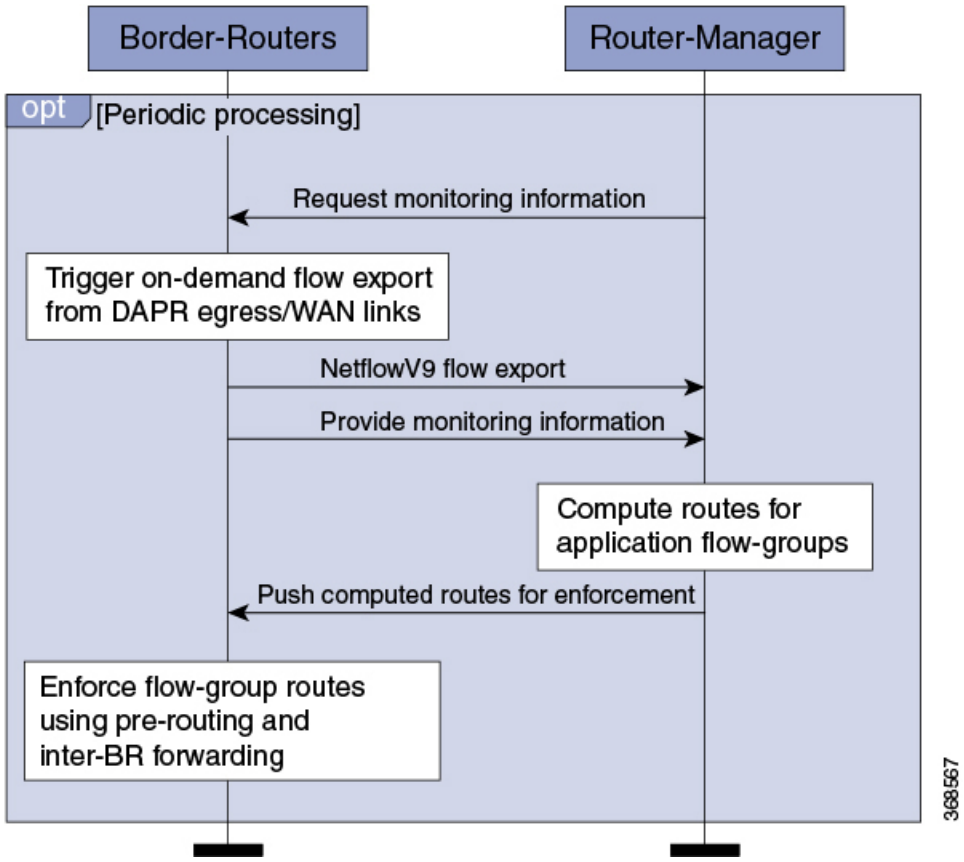
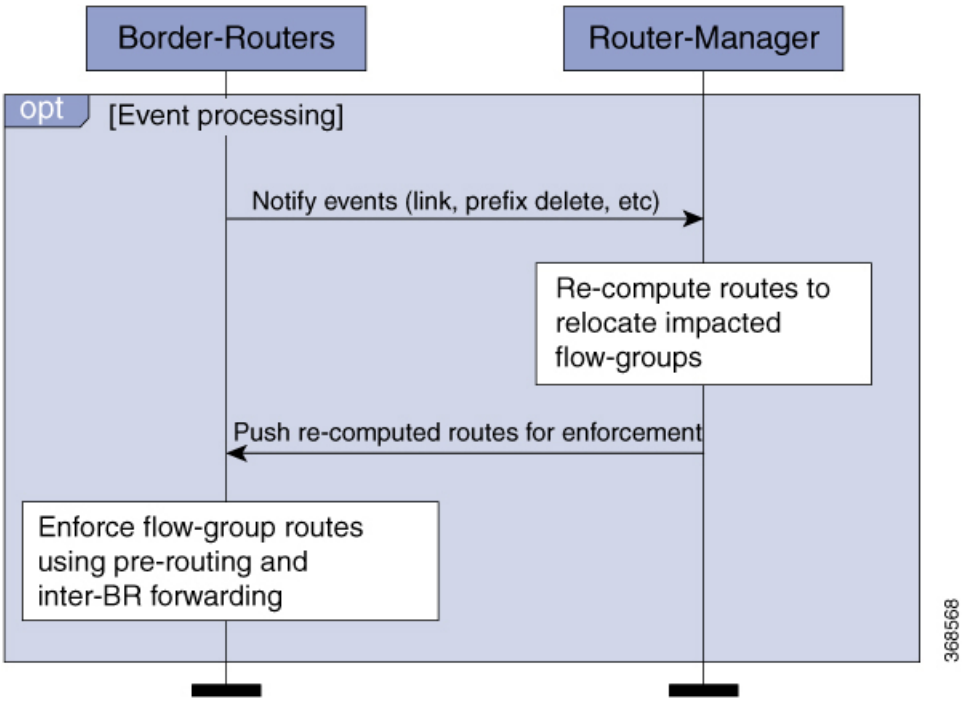


Figure 22: DAPR Event Processing

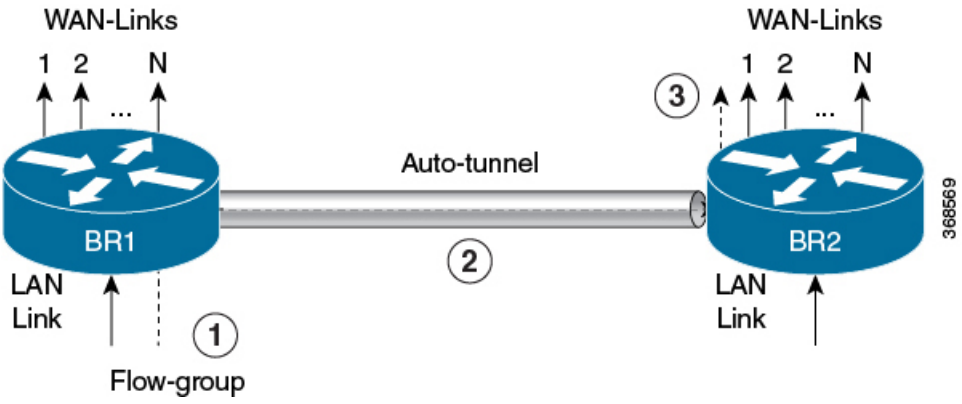


Inter BR Forwarding

BRs create IP/GRE tunnels (referred to as auto-tunnels) to neighbor-BRs learnt from the RM. The inter-BR auto-tunnels are between the BR loopback IP addresses.

With site-wide policy routing, ingress BR for a flow-group and the egress BR can be different and this requires forwarding of traffic between BRs. DAPR uses auto-tunnels for loop-free forwarding of traffic between BRs.

Figure 23: Auto-tunnel based Inter-BR Forwarding



DAPR Operations

DAPR operation is based on three key building blocks:

- Monitoring
- Flow Route Computation
- Flow Route Enforcement

Monitoring

DAPR monitoring involves BRs monitoring and exporting the following information to RM for the flow route computation based on the site-wide visibility:

- Bandwidth and utilization of DAPR egress interfaces (DAPR-enabled WAN links)
- Prefixes learned through the DAPR egress interfaces
- Application flow-groups egressing the DAPR egress interfaces
- Inter-BR availability through the auto-tunnels

Flow Route Computation

Flow Route Computation Logic:

Invokes DAPR RM route-compute logic to compute routes for newly discovered flow-groups. It also re-computes routes for existing flow-groups to re-locate either due to events impacting current routes or current routes being not the best routes. Invokes route-compute on a per flow-group basis and involves following steps:

1. Create a list of viable egress interfaces that meet all the following criteria.
    - Egress interface has the flow destination availability.
    - Egress interface bandwidth is above the specified minimum-bandwidth.
    - Egress interfaces have the headroom for the flow.
    - Egress BR has the bidirectional inter-BR reachability to ingress-BR.
2. Select the best egress interface which is based on the following parameters as tie breakers:
    - Egress that has the higher specified preference for the flow-group.
    - Egress that has higher projected percentage-headroom (projected remaining link utilization).
    - Egress that has the lesser number of flows.
    - Egress link stickiness.

Flow-group Selection Logic for Re-location:

When an egress interface exceeds the specified link thresholds, some of the flow-groups re-locates to other egress interfaces. Flow-groups are selected in the following order for re-location:

- Flow-groups that have no preference for the current egress interface (pref-level = none).
- Flow-groups for which the current egress interface has third preference (pref-level = 3).
- Flow-groups for which the current egress interface has second preference (pref-level = 2).

- Flow-groups for which the current egress interface has first preference (pref-level = 1).
- If there are multiple flow-groups that have the same preference level for the current egress, any of the flow-groups can be selected for the re-location (indeterminate).

Flow States

The following table lists the DAPR flow-group states:

Table 29: DAPR flow-group States

State Transition	Description
Unmanaged (U)	Newly discovered flow-group by RM.
Managed (M)	• For the flow-group with preference policy, flow-group assigned to its most preferred interface • For the flow-group with no preference policy, flow-group assigned to any viable interface
Out-of-policy (O)	• For the flow-group with preference policy, flow-group assigned to its lesser/non-preferred interface. • For the flow-group with no preference policy - NA.
Deleted (D)	Flow-group that was in M/O state and is marked for deletion.

The following lists lifecycle of a flow-group that does not have a preference policy.

State Transition	Description
U ⇨ M	Flow-group assigned to any viable egress
U ⇨ D	• Flow-group discovered from non-DAPR ingress • Flow-group discovered from multiple BRs/egresses • No viable egress available for the flow-group
M ⇨ M	Flow-group relocated due to events
M ⇨ D	• Flow-group expiry - not seen for multiple cycles • Flow-group discovered from invalid egress/ingress • Flow-group could not be relocated as part of event processing

The following lists the lifecycle of a flow-group that has a preference policy.

State Transition	Description
U ⇨ M	Flow-group assigned to its most preferred egress
U ⇨ O	Flow-group assigned to lesser or non-preferred egress
U ⇨ D	• Flow-group discovered from non-DAPR ingress • Flow-group discovered from multiple BRs/egresses • No viable egress available for the flow-group
M ⇨ O	Flow-group re-located to lesser/non-preferred egress as part of event processing.
O ⇨ M	Flow-group relocated to its most preferred egress as part of event or periodic OOP flow processing.
O ⇨ O	Flow-group re-located to lesser/non-preferred egress as part of event or periodic OOP flow processing.
M ⇨ M	Flow re-located to another most-preferred egress as part of processing an event where current egress is no longer viable.
M/O ⇨ D	• Flow-group expiry that is not seen for multiple cycle. • Flow-group discovered from invalid egress or ingress. • Flow-group that are part of event processing cannot be relocated.

Flow Route Enforcement

Flow-group route enforcement involves the following steps:

1. RM pushes the computed route for a flow-group to its ingress-BR. For example, the BR that is currently receiving this flow-group from LAN. The flow-group route consist of (Egress-BR, Egress-interface, Next-hop-IP).
2. Ingress BR enforces the flow-group route as follows:
   - If the egress BR is same as the ingress BR, pre-routing bypasses the routing.
   - If the egress BR is not same as ingress BR, pre-routing forwards traffic to egress BR over the auto-tunnel. The auto-tunnel carries metadata specifying the egress interface to use on the egress-BR.

DAPR Features

DAPR supports the following key features:

1. Link preference
2. Link load balancing
3. Application flow-group whitelisting
4. RM redundancy

Link Preference

This feature ensures application performance by dynamically steering application flows to the specified preferred WAN links.

Link Load Balancing

This feature ensures uniform utilization of the DAPR-enabled WAN links by dynamically steering application flows across WAN links based on changing link bandwidth or utilization and flow rates.

Application Flow-group Whitelisting

This feature allows flow-groups to skip DAPR action. Such flows take the path as determined by regular routing and are not managed by DAPR. Currently, the whitelisted flow-groups are reported by BRs to RM but are ignored by RM.

One of the use cases where this feature is useful is for DAPR to bypass and not manage traffic that is required for its operation such as routing protocol traffic.

RM Redundancy

DAPR supports stateless RM redundancy using anycast-IP with no state synchronization between the RMs. In case the current RM goes down or becomes unreachable, the TCP control connection keepalives detect this and reset the connection, and the new connection goes to the other RM.

Like with any other anycast based redundant setup, routing must be setup to ensure that only one of the RMs is reachable from all the BRs at any time.

DAPR Scalability and Responsiveness

DAPR supports the following scaling numbers:

Table 30: Standalone RM and BR

RM Scale		
Description	**Scaling Numbers: Cisco IOS XE Release 16.11.1**	**Scaling Numbers: Cisco IOS XE Release 17.3.1 Onwards**
Maximum number of BRs	20	40
Maximum number of WAN links per BR	20	60

RM Scale		
Description	**Scaling Numbers: Cisco IOS XE Release 16.11.1**	**Scaling Numbers: Cisco IOS XE Release 17.3.1 Onwards**
Maximum number of WAN links across all BRs	400	2400
Maximum number of destination prefixes	525/2100	2100/8400
Maximum number of application flow-groups	33,600	33,600
BR Scale		
Maximum number of destination prefixes	175/700	420/1680
Maximum number of application flow-groups	11,200	6,720

Table 31: Co-located RM and BR Scale

Description	Scaling Numbers: Cisco IOS XE Release 16.11.1	Scaling Numbers: Cisco IOS XE Release 17.3.1 Onwards
Maximum number of BRs	1	1
Maximum number of WAN links per BR	8	8
Maximum number of WAN links across all BRs	8	8
Maximum number of destination prefixes/routes	35/140	14/56
Maximum number of application flow-groups	3600	1,344

DAPR Responsiveness

The DAPR responsive time includes:

1. DAPR response-time to critical events = ~5 seconds.
    - WAN link down, route deletion, WAN link hard threshold exceed
2. DAPR response-time to non-critical events = ~30 seconds
    - WAN link soft threshold exceed, out-of-policy flows.

Benefits of DAPR

DAPR offers the following benefits compared to other solutions:

1. DAPR has no overlay dependency: DAPR does not require an overlay and it can manage the overlay or underlay traffic.
2. Synchronized and predictable system: RM performs a synchronized collection of monitoring information from all the BRs. RM performs the flow route computation and route push at designated periodic that intervals based on the latest monitoring information. BRs use an on-demand flow export that is triggered by periodic requests from the RM for the synchronized flow export from all the BRs.
3. Predictable route enforcement: DAPR uses policy routing (PBR) on the BRs to enforce flow routes from the RM. BRs use PBR batching feature to push the updated flow routes that are received from the RM to the data plane. This avoids chattiness between the control and data plane, and ensures predictable dynamic flow route enforcement.
4. Inter-BR availability tracking: DAPR monitors the state of the auto-tunnels and thus the reachability between BRs. RM maintains the inter-BR reachability matrix and uses it for the route computation.
5. Simplified forwarding state distribution: RM pushes the flow routes only to the ingress-BR. Ingress-BR enforces the flow routes using policy routing (PBR) and inter-BR forwarding over auto-tunnels for the route enforcement.
6. Loop-free inter-BR forwarding: Forwarding of the inter-BR traffic over auto-tunnels ensures that traffic does not loop between BRs.
7. No restriction that BRs must be a L2-adjacent: The inter-BR IP or GRE auto-tunnels remove the restriction that BRs at a site be L2 adjacent.
8. Inter-BR resiliency with multiple LANs: The inter-BR auto-tunnels provide the resiliency when BRs are interconnected over multiple LANs.
9. Supports variable-BW Radio WAN links.
10. Supports virtual-access interfaces as WAN interfaces.
11. Simplified and reduced configuration: DAPR has simplified and reduced configuration by avoiding any BR-specific configuration on the RM.

Prerequisites for DAPR Solution

To configure the DAPR solution:

1. Configure DAPR RM and BRs with a loopback interface with a host IP address.
   - Use the RM or BR loopback IPs for RM-BR control communications, and for the inter-BR auto-tunnels.
2. RM-BR availability (between RM and BR loopback IPs).

- RM is purely a control plane entity and does not participate in data plane forwarding. Therefore, keep the availability between BRs and RM separate from the BR availability to remote-sites. In other words, do not extend the BR WAN-side routing to RM, which would load the RM unnecessarily.
- We recommended to use either a separate routing protocol instance between BR and RMs or static routes.
- RM must not be reachable from the BRs through DAPR egresses.

3. Inter-BR availability (IP or GRE auto-tunnels between BR loopback IPs).
   - Like BR-RM availability, it is preferable to keep the inter-BR availability separate from the BR availability to remote-sites.
   - As the DAPR tracks the inter-BR availability (and the auto-tunnel UP/DOWN status) and uses this in route computations, it is recommended to use dynamic routing protocol instead of static routes for availability between BR loopbacks.
   - If the RM-BR availability is using a separate routing protocol instance, use the same instance for inter-BR loopback availability as well.
   - Inter-BR availability must NOT be through DAPR egresses.
   - Avoid static routes for inter-BR availability, as there are no tunnel keepalives to monitor availability.
4. All possible paths (not just the best path) to remote sites that are reachable through DAPR egress interfaces (DAPR-enabled WAN links) must be available in the routing table either as equal cost or unequal cost routes. This requires tuning of routing protocols metrics.

Restrictions for DAPR

The following restrictions apply to DAPR:

- DAPR supports only IPv4.
- DAPR is supported on RAR and PPPoE interfaces only in RAR bypass mode.
- DAPR identifies application flow groups that are based on a 3-tuple of {source IP-address, destination IP-address, DSCP} where the source and destination IP addresses are host addresses. This means DAPR flow-group currently consists of a single flow with a unique source-IP, destination-IP, and DSCP value.
- DAPR does not support identification of application flow groups using NBAR or 5-tuple (source-prefix, destination-prefix, protocol, source-ports, destination-ports).
- DAPR does not use probes and hence does not support monitoring of delay, jitter, and packet loss on WAN links.

Supported Platforms for DAPR

The following table provides the supported platforms for DAPR.

Table 32: Supported Platforms for DAPR

DAPR Components	Cisco 4000 Series ISR with Cisco IOS-XE Release 17.3.1 Onwards	Cisco ASR 1000 with Cisco IOS XE Release 17.3.1 Onwards	Cisco CSR 1000v with Cisco IOS XE Release 17.3.1 Onwards	ISRv with Cisco IOS XE Release 17.3.1 Onwards
Route-Manager (RM)	Yes	Yes	Yes	No
Border-Router (BR)	Yes	Yes	Yes	Yes
Co-located BR and RM	Yes	Yes	Yes	Yes

Note DAPR is supported only on Cisco 4451, 4300 ISR, and ASR 1001-X routers.

How to Configure DAPR

To configure DAPR, follow these steps:

1. Configure the loopback interfaces on BRs and RM.
   - Establish the RM-BR reachability between BR and RM loopbacks.
   - Establish the inter-BR reachability between BR loopbacks.
2. Ensure that all paths to remote destinations are in the routing table (RIB).
3. Configure the RM.
4. Configure the BR.

Configuring DAPR instance

DAPR instance is a container for DAPR RM and/or BR configuration. Currently, only a single DAPR instance is supported. DAPR instance is identified by a user-defined string or by the string *default*.

Note There are multiple instances where the interface utilization or bandwidth may be inaccurate. This can cause undesirable Traffic Class movements even for very small changes (or inaccuracies). To avoid the undesirable flow movements, route-manager allows 5% margin in inaccuracies and to flow stickiness even when there are changes upto 5%.

```
Device(config)#?
  Dapr    Dynamic Application Policy Routing (DAPR)
               configuration

DAPR(config)#dapr ?
```

```
  WORD     Instance Name
  default    Default DAPR Instance

Device(config)#dapr default
DAPR(config-dapr-instance)#
DAPR(config)#dapr dapr-instance-1
 DAPR instance 'default' exits. Single instance allowed.

Device(config-dapr-instance)#?
DAPR Instance Configurations commands:
  border-router  DAPR border router (BR) configuration
  route-manager  DAPR route manager (RM) configuration
```

Configuring Route Manager

Configure the DAPR RM within the DAPR instance as show in this example:

```
Device(config-dapr-instance)#route-manager
Device(config-dapr-route-manager)#?
Router manager configuration commands:
  authentication    Authentication parameters
  border-routers    Authorized border routers
  class             Application class parameters
  link-thresholds   BR egress link thresholds
  shutdown          Disable route manager instance
  source-interface  Route manager address source
```

Shutdown the RM before creating or modifying any RM configuration.

```
Device(config-dapr-route-manager)#link-thresholds
 RM should be in shutdown mode for any config change

Device(config-dapr-route-manager)#shutdown
 %DAPR_RM-5-RM_STATUS: Shutdown
 %DAPR_RM-5-RM_STATUS: Inactive

Device(config-dapr-route-manager)#link-thresholds
Device(config-dapr-rm-link-thresholds)#

Device(config-dapr-route-manager)#no shutdown
%DAPR_RM-5-RM_STATUS: Active
```

Configure the following mandatory parameters to RM to start listening to BR connections:

- RM source interface (loopback interface) with a valid IP-address
- Authentication password
- List of authorized BRs, with at least one entry

```
Device#show running-config | section dapr
dapr default
 route-manager
 ! Config incomplete
```

Configuring the RM Source Interface

RM uses the source interface IP address for control communication with BRs. RM source interface can only be a loopback interface.

```
Device(config-dapr-route-manager)#?
Router manager configuration commands:
```

```
  source-interface  Route manager address source
Device(config-dapr-route-manager)#source-interface ?
  Loopback  Loopback interface
```

Example

```
dapr default
 route-manager
  source-interface Loopback0
interface Loopback0
 description RM-loopback
 ip address 11.0.0.1 255.255.255.255
```

Configuring DAPR Authentication

RM uses passwords to authenticate BRs. Note that DAPR authentication is unidirectional in that it is only for BR authentication to RM and not vice versa. The password is carried in plaintext over the BR-RM TCP-based control connection.

Use IKE/IPsec for more secure and mutual authentication of RM and BRs. For more information, see the IOS IKE/IPsec configuration guide for configuring IKE/IPsec.

DAPR authentication is a mandatory configuration.

```
Device(config-dapr-route-manager)#?
Router manager configuration commands:
  authentication    Authentication parameters

Device(config-dapr-route-manager)#authentication ?
  password  assign password (Max of 25 characters)
Device(config-dapr-route-manager)#authentication password ?
  0     Specifies an UNENCRYPTED password will follow
  4     Specifies an SHA256 HASHED password will follow
  LINE  The UNENCRYPTED (cleartext) 'password' string
```

Note that even if the authentication password is entered in plaintext, encrypted password is displayed in the running-config.

```
Device(config-dapr-route-manager)#authentication password dapr123
Device#show running-config | section dapr
dapr default
 route-manager
  authentication password 4 U28mHpS4suXM7r6q3U3E.oDXKCESijH3TSF6FHKrYHA
```

Example

```
dapr default
 route-manager
  authentication password 4 U28mHpS4suXM7r6q3U3E.oDXKCESijH3TSF6FHKrYHA
```

Configuring DAPR Authorization

DAPR authorization consists of a list of BR IP addresses that are authorized to register with the RM. The list can have a maximum of 20 entries for a standalone RM and a single entry for a co- located RM and BR. You must configure DAPR authorization with at least one entry.

```
Devic(config-dapr-route-manager)#?
Router manager configuration commands:
  border-routers    Authorized border routers

Device(config-dapr-route-manager)#border-routers ?
  <cr>
```

```
Device(config-dapr-rm-brs)#?
RM border router configuration commands:
  A.B.C.D  Border router address
```

Example

```
dapr default
 route-manager
  border-routers
   10.0.0.2
```

Configuring DAPR Thresholds

DAPR thresholds specify the thresholds for DAPR egress interfaces on the BRs. RM pushes the thresholds to BRs in the registration response on a successful registration. BRs enforce the thresholds by monitoring the DAPR egress interfaces and reporting any threshold violation to the RM. RM re-computes routes in order to relocate the application flow groups impacted by the threshold violations.

Following are the currently supported thresholds:

- Minimum bandwidth - Specifies the minimum bandwidth (in kbps) in order for DAPR egress interfaces to be considered viable and used in route computations. The default value is 500kbps.
- Maximum percent utilization - Specifies the maximum utilization (in percentage) beyond which DAPR egress interfaces would be considered out-of-policy. The default value is 50%.
- Configuring DAPR thresholds is optional and there are default values for thresholds.

```
Devcie(config-dapr-route-manager)#?
Router manager configuration commands:
  link-thresholds   BR egress link thresholds

Device(config-dapr-route-manager)#?
Router manager configuration commands:
  class             Application class parameters

Device(config-dapr-route-manager)#link-thresholds
Device(config-dapr-rm-link-thresholds)#?
RM link threshold configuration commands:
  max-utilization  Maximum % utilization (default = 50)
  min-bandwidth    Minimum bandwidth (kbps) for viability (default = 500)
```

Example

```
dapr default
 route-manager
  link-thresholds
   max-utilization 50
   min-bandwidth 500
```

Configuring DAPR Preference Policy

DAPR preference policy allows specifying a list of preferred links for a set of flow-groups. DAPR preference policy is an ordered sequence of DAPR application classes. Each class specifies match criteria for flow-groups using an access-list and the first, second and third preferred link-groups. .

Link-group is an arbitrary group of DAPR egress interfaces that is referenced in preference policy. Configure link-group membership on the BR egress interfaces. BRs communicate the membership information to RM in the registration request. A DAPR egress interface can be part of a single link-group.

DAPR application classes are processed in the order of class sequence number and first match is used. Up to 255 classes can be configured. Each class must have a unique combination of class name and sequence number. Configuring DAPR preference policy is optional.

```
Device(config-dapr-route-manager)#?
Router manager configuration commands:
  class             Application class parameters

Device(config-dapr-route-manager)#class ?
  WORD  Application class name
```

Up to 255 application classes can be configured.

```
Device(config-dapr-route-manager)#class class1 ?
  <1-255>  Application class processing sequence

Device(config-dapr-route-manager)#class class1 1 ?
  <cr>  <cr>
```

Each class must have a unique combination of class name and sequence number.

```
Device(config-dapr-route-manager)#class class2 1
 Class 'class1 1' exists.
 Changing class name or sequence number not allowed.

Device(config-dapr-route-manager)#class class1 2
 Class 'class1 1' exists.
 Changing class name or sequence number not allowed.

Device(config-dapr-rm-class)#?
RM application class configuration commands:
  match            Match criteria
  path-preference  Specify path preference
```

Application flow-group matching is based on extended ACL and using only source, destination and dscp.

```
Device(config-dapr-rm-class)#match ?
  access-list  Specify access-list

Device(config-dapr-rm-class)#match access-list ?
  WORD  IP Named Extended Access list name

Device(config-dapr-rm-class)#match access-list access-list1
 Note: DAPR Flow match based on source, destination and dscp only.
       Other ACL fields ignored.
Device(config-dapr-rm-class)#
```

Up to 3 link-groups can be specified as path preference.

```
Device(config-dapr-rm-class)#path-preference
Device(config-dapr-rm-class-path-pref)#?
RM class path preference configuration commands:
  <1-255>  Path preference sequence number

Device(config-dapr-rm-class-path-pref)#1 ?
  WORD  Link group name (max 50 characters)

Device(config-dapr-rm-class-path-pref)#1 link-group1
Device(config-dapr-rm-class-path-pref)#2 link-group2
Device(config-dapr-rm-class-path-pref)#3 link-group3
Device(config-dapr-rm-class-path-pref)#4 link-group4
 Max 3 path preferences allowed in a class.
```

Example

```
dapr default
 route-manager
  class class1 1
   match access-list access-list1
   path-preference
    1 link-group1
    2 link-group2
    3 link-group3

ip access-list extended access-list1
 permit ip any any
```

Configuring DAPR Whitelisting

DAPR whitelisting policy allows specifying a set of flow-groups egressing DAPR egress interfaces that must not be managed by DAPR. Such flow-groups would take regular routing paths.

DAPR whitelist policy can be configured using a DAPR application class of type *bypass*. The bypass application class specifies :

- A match criteria for flow-groups using an access-list. (Optional) You can configure only a single DAPR whitelist policy.
- Minimum flow bandwidth for flow admission. If present, flows having bandwidth below the specified value are ignored by DAPR. This configuration is optional and by default all flows are managed by DAPR.

```
Device(config-dapr-route-manager)#class ?
  WORD  Application class name

Device(config-dapr-route-manager)#class bypass_class ?

  <1-255>  Application class processing sequence
  type     Application class type

Device(config-dapr-route-manager)#class bypass_class type ?
  bypass  Application class type bypass

Device(config-dapr-route-manager)#class bypass_class type bypass

RM application class configuration commands:
  exit           Exit from RM class configuration submode
  match          Match criteria
  min-flow-rate  Minimum bandwidth (kbps) for flow admission
  no             Negate or set default values of a command.

evice(config-dapr-rm-class)#match ?

  access-list  Specify access-list

Device(config-dapr-rm-class)#match access-list ?
  WORD  IP Named Extended Access list name

Device(config-dapr-rm-class)#match access-list bypass-acl
Note: DAPR Flow match based on source, destination and dscp only. Other ACL fields ignored.


Device(config-dapr-rm-class)#min-flow-rate 5000.

Example
dapr default

route-manager
```

```
  class bypass_class type bypass
   match access-list bypass-acl
   min-flow-rate 5000

ip access-list extended bypass-acl
permit ip any any dscp ef
```

Verifying RM

Verify RM configuration and operation using the following show commands.

```
Device#show dapr route-manager ?
  border-router    Border router information
  flow-groups      Flow-group learnt from BRs
  link-groups      Link-group membership information
  route-table      Prefixes/routes learnt from BRs
  summary          RM Summary information

Device#show dapr route-manager border-router ?
  A.B.C.D    BR address
  neighbors  BR neighbor connectivity information
  summary    BR summary information
  |        Output modifiers
  <cr>       <cr>

Device#show dapr route-manager link-groups ?
  WORD  link-group name
  |        Output modifiers
  <cr>  <cr>

Device#show dapr route-manager route-table ?
  A.B.C.D  BR address - routes learnt from this BR
  |        Output modifiers
  <cr>     <cr>

Device#show dapr route-manager flow-groups ?
  detail      flow-groups detail
  egress-br   flow-groups ingressing this BR
  ingress-br  flow-group ingressing this BR
  match       flow-group match criteria
  |           Output modifiers
  <cr>        <cr>

Device#show dapr route-manager flow-groups match ?
  destination  flow-groups matching this destination prefix
  dscp         flow-groups matching this dscp
  source       flow-groups matching this source prefix
  |            Output modifiers
  <cr>         <cr>
```

Configuring Border Router

DAPR BR is configured under DAPR instance.

```
Device(config-dapr-instance)#border-router
Device(config-dapr-border-router)#?
Border router configuration commands:
  authentication    Authentication parameters
  route-manager     Route manager address
```

```
  shutdown          Disable border router instance
  source-interface  Border router address source
```

Shutdown BR before creating or modifying any BR configuration.

```
Device(config-dapr-border-router)#source-interface loopback 1
 BR should be in shutdown mode for any config change

Devcie(config-dapr-border-router)#shutdown
%DAPR_BR-5-STATUS: shutdown

Device(config-dapr-border-router)#source-interface loopback 1
Device(config-dapr-border-router)#no shutdown
```

```
Device#show running-config | section dapr
dapr default
 border-router
  ! Config incomplete
```

DAPR BR Mandatory Configuration

Configure the BR with the following mandatory parameters for a BR to start TCP control connection and registration with RM

- BR source interface (loopback interface) with a valid IP-address.
- Authentication password.
- RM IP address (must be reachable through non DAPR-egress interfaces).
- At least one interface configured as DAPR egress.

```
Device#show running-config | section dapr
dapr default
 border-router
  ! Config incomplete
```

Configuring the BR Source Interface

BRs use the source interface IP address for control communication with RM as well as for the inter-BR auto-tunnels(IP/GRE). RM source interface can only be a loopback interface. Configuring BR source interface is mandatory.

```
Device(config-dapr-route-manager)#?
Router manager configuration commands:
  source-interface  Route manager address source

Device(config-dapr-route-manager)#source-interface ?
  Loopback  Loopback interface
```

Example

```
dapr default
 border-router
  source-interface Loopback0

interface Loopback0
 description BR-loopback
 ip address 10.0.0.1 255.255.255.255
```

Configuring DAPR Authentication

BRs use passwords to authenticate to RM. Note that DAPR authentication is unidirectional in that it is only for BR authentication to RM and not vice versa. The password is carried in plain text over the BR-RM TCP-based control connection.

Use IKE/IPsec for more secure and mutual authentication of RM and BRs. For more information, see the IOS IKE/IPsec configuration guide for configuring IKE/IPsec.

DAPR authentication is a mandatory configuration.

```
Device(config-dapr-border-router)#?
Border router configuration commands:
  authentication    Authentication parameters
  route-manager     Route manager address
  shutdown          Disable border router instance
  source-interface  Border router address source

Device(config-dapr-border-router)#authentication ?
  password  Specify the password (Max of 25 characters)

Device(config-dapr-border-router)#authentication password ?
  0     Specifies an UNENCRYPTED password will follow
  4     Specifies an SHA256 HASHED password will follow
  LINE  The UNENCRYPTED (cleartext) 'password' string
```

Note that even if the authentication password is entered in plaintext, encrypted password is displayed in the running-config.

```
Device(config-dapr-border-router)#authentication password dapr123
Device#show running-config | section dapr
dapr default
 border-router
  authentication password 4 U28mHpS4suXM7r6q3U3E.oDXKCESijH3TSF6FHKrYHA
```

Example

```
dapr default
 border-router
  authentication password 4 U28mHpS4suXM7r6q3U3E.oDXKCESijH3TSF6FHKrYHA
```

Configuring DAPR Egress Interfaces and Link-group Membership

Configure at least one interface (WAN facing interface) as a DAPR egress interface. This is required for a BR to start initiating TCP connection and registration to RM. DAPR manages only the flow-groups egressing DAPR egress interfaces.

Optionally configure a DAPR egress interface with link-group membership. A DAPR egress interface can only be part of a single link-group. BR reports DAPR egress interfaces along with any link-group membership information to the RM in registration request.

DAPR egress and link group membership can only be configured on the following interfaces types:

- PPPoE/RAR virtual-template interface
- PPPoE/RAR virtual-access interface
- Serial interface
- Ethernet main and sub-interface

Note An interface can be configured as either DAPR egress or ingress but you cannot configure not both.

Configuring at least one DAPR egress interface is mandatory. Configuring link-group membership is optional.

```
Devcie(config)#interface Loopback 0
Device(config-if)#dapr ?
  egress   dapr egress interface
  ingress  dapr ingress interface
Device(config-if)#dapr egress
% ERROR: Interface not supported as DAPR Egress

Device(config)#interface Serial2/00
Device(config-if)#dapr ?
  egress   dapr egress interface
  ingress  dapr ingress interface

Device(config-if)#dapr egress ?
  link-group  specify link group name (max 50 characters)
  <cr>        <cr>

Device(config-if)#dapr egress link-group ?
  WORD  link group name

Device(config-if)#dapr egress link-group LG1
```

Example

```
interface Serial2/0
 dapr egress link-group LG2
```

Configuring DAPR Ingress Interfaces

At least one interface (LAN facing interface) must be configured as a DAPR ingress interface. Configuring DAPR ingress interface is not mandatory for a BR to start registration. However, only the flow-groups entering a BR through DAPR ingress interfaces (DAPR-enabled LAN interfaces) are managed by DAPR. .

Note An interface can be configured as either DAPR egress or ingress but not both.

DAPR ingress only be configured on Ethernet main and sub-interfaces.

```
Device(config)# interface Loopback 0
Device(config-if)#dapr ingress
% ERROR: Interface not supported as DAPR Ingress

Device(config)# interface Ethernet0/0
Device(config-if)#dapr ingress
```

Example

```
interface Ethernet0/0
 dapr ingress
```

Verifying BR

Verify BR configuration and operation using the following show commands:

```
Device#show dapr border-router ?
  interfaces  BR interface information
  neighbors   BR neighbor information
  summary     BR status information

Device#show dapr border-router neighbors ?
  |     Output modifiers
  <cr>  <cr>

Device#show dapr border-router interfaces ?
  metrics  Egress interface metrics
  |        Output modifiers
  <cr>     <cr>
```

Configuring DAPR Co-located RM and BR

DAPR RM and BRs would be commonly configured on separate routers. For single edge router sites, RM and BR can be configured on the same router under the same DAPR instance, which is referred to as co-located RM/BR.

Following restrictions apply to co-located RM/BR:

- Co-located RM and BR must use different source interfaces (different loopback interfaces).
- Co-located RM supports a single BR.
- Co-located RM does not support external BRs.
- Co-located BR supports a maximum of 8 DAPR egress interfaces and 3360 flow-groups.

DAPR Yang Model

YANG data model is defined for DAPR feature which allows user to add, modify, and delete configuration programmatically using NETCONF.

To make any programmatical changes, use the **shutdown** RPC command first and followed by configuration changes including **no shutdown** command. Operational yang model is currently not supported.

Troubleshooting DAPR

To troubleshoot the DAPR configuration, use the debug commands or the syslog messages.

DAPR RM and BR Syslogs

The following table provide the syslog for RM and BR:

Table 33: RM Syslog

Syslog	Severity Level	Description
BR_REG_FAILED	Error(3)	BR Registration failed
BR_RESET	Error(3)	RM reset the BR

Syslog	Severity Level	Description
FLOW_EXP_PKTS_MISSED	Error(3)	Flow export packets missed
FLOW_INVALID_EGRESS	Error(3)	Flow discovered from unexpected egress
APP_RT_COMPUTE_FAILED	Error(3)	App route compute failed for flow-group
NO_VIABLE_PATH	Warning(4)	No viable path found for flow-group
APP_REROUTE_FAILED	Warning(4)	App route re-compute failed for flow-group
FLOW_EXP_PKT_INVALID_SEQ	Warning(4)	Unexpected sequence number in flow export packet
FLOW_DATA_RECS_IGNORED	Warning(4)	Flow data records ignored
FLOW_INVALID_INGRESS	Warning(4)	Flow discovered from unexpected ingress
FLOW_MULTI_EGRESS	Warning(4)	New flow discovered from multiple egresses
INTERNAL_ERROR	Warning(4)	Internal error
RIB_MISMATCH	Warning(4)	Mismatch of RIB database between BRs and RM
BR_STATUS	Notification(5)	Border-Router status on RM
RM_STATUS	Notification(5)	RM status changed
APP_RT_INSTALL	Informational(6)	App route installed for flow-group
APP_RT_DEL	Informational(6)	App route deleted for flow-group
BR_EVENT	Informational(6)	RM received event from BR
RM_RESET	Informational(6)	RM reset

Table 34: DAPR BR Syslogs

Syslog	Severity Level	Description
PREFIX_LIMIT_EXCEEDED	Warning(4)	DAPR RIB prefixes exceeded
FLOW_LIMIT_EXCEEDED	Warning(4)	DAPR Flows exceeded
RMAP_LIMIT_EXCEEDED	Warning(4)	DAPR route-map entries exceeded max allowed

Syslog	Severity Level	Description
INTERNAL_ERROR	Warning(4)	Internal error
STATUS	Notification(5)	BR status changed
RESET	Notification(5)	Border-Router reset
RM_ROUTE_INVALID	Notification(5)	Invalid route from BR to RM
NBR_ROUTE_INVALID	Notification(5)	Invalid route to neighbor BR
NBR_TUNNEL_UPDOWN	Notification(5)	Status of tunnel to neighbor BR changed
EGRESS_INTF_THRESHOLD_EXCEED	Notification(5)	DAPR egress interface utilization threshold exceeded
EGRESS_INTF_NOT_VIABLE	Notification(5)	DAPR egress interface not viable
EGRESS_INTF_UPDOWN	Notification(5)	DAPR egress interface status
INGRESS_INTF_UPDOWN	Notification(5)	DAPR ingress interface status

Debug Commands

The following are the DAPR debug commands:

```
Device#debug ?
  dapr                     Enable Dapr debugs

Device#debug dapr ?
  border-router  Enable Border Router debugs
  packet         Enable Packet debugs
  route-manager  Enable Route Manager debugs
  socket         Enable Socket debugs

Device#debug dapr route-manager ?
  all             Enable RM RIB/Flow-Collector/Route-Compute/Events debugging
  events          Enable RM Events debugging
  flow-collector  Enable RM Flow-Collector debugging
  rib             Enable RM RIB debugging
  route-compute   Enable RM Route-Compute debugging

Devie#debug dapr border-router ?
  all          Enable BR RIB/Flow-Export/Flow-Route/Inter-BR/Wan-Metric/Events
               debugging
  events       Enable BR Events debugging
  flow-export  Enable BR Flow-Export debugging
  flow-route   Enable BR Flow-Route debugging
  inter-br     Enable BR Inter-BR Tunnel debugging
  rib          Enable BR RIB debugging
  wan-metric   Enable BR Wan-Metric debugging

Device#debug dapr packet ?
  detail  Enable Packet detail debugging
  dump    Enable Packet dump debugging
  error   Enable Packet error debugging
  <cr>    <cr>
```

```
Device#debug dapr socket ?
  detail  Enable Socket detail debugging
  error   Enable Socket error debugging
  <cr>    <cr>
```

DAPR Conditional Debug Commands

Conditional debug commands are supported only on RM.

```
Device#debug dapr route-manager ?
  condition       Enable RM Conditional debugging
```

Conditional debugging can be based on BR IP address and the flow-group parameters.

```
Device#debug dapr route-manager condition ?
  br-ip        Enable RM Condition based on the BR ip address
  flow-groups  Flow-group learnt from BRs
  unmatched    Output debugs even if no context available

Device#debug dapr route-manager condition flow-groups ?
  destination  flow-groups matching this destination prefix
  dscp         flow-groups matching this dscp
  egress-br    flow-groups egressing this BR
  ingress-br   flow-group ingressing this BR
  source       flow-groups matching this source prefix
  <cr>         <cr>
```

DAPR conditional debugging status can be checked using the below command.

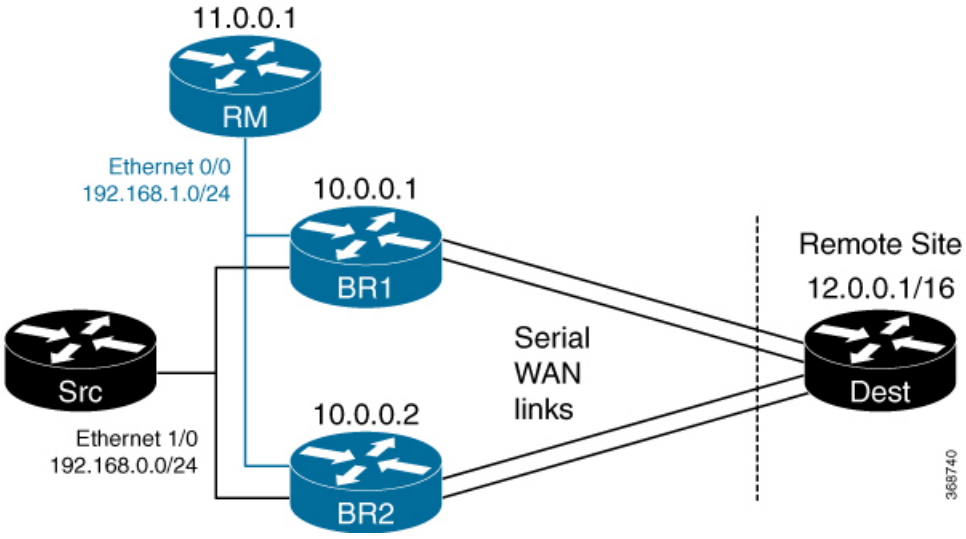
```
Device#show dapr route-manager debug-condition
BR addresses under debug are:
10.0.0.1,
Flow-groups under debug are(SRC(mask)/DST(mask)/DSCP/Egress/Ingress):

DAPR RM Conditional debug context unmatched flag: OFF
Device#
```

Configuration Examples

Example for DAPR Standalone RM and BR

This configuration example is based on a sample DAPR topology shown in the figure below. The topology consists of a standalone RM, 3 BRs, traffic source, and destination.

Figure 24: DAPR Topology



Configuring Route-Manager

The following example shows how to configure a RM:

```
dapr default
 route-manager
  source-interface Loopback0
  authentication password 4 tnhtc92DXBhelxjYk8LWJrPV36S2i4ntXrpb4RFmfqY
  link-thresholds
   max-utilization 50
   min-bandwidth 500
  border-routers
   10.0.0.2
   10.0.0.1
  class whitelist type bypass
   match access-list access-list2
  class class1 1
   match access-list access-list1
   path-preference
    10 LG1
    20 LG2
!
interface Loopback0
 description RM-loopback
 ip address 11.0.0.1 255.255.255.255
!
interface Ethernet0/0
 description RM-BR LAN
 ip address 192.168.0.1 255.255.255.0
!
ip route 10.0.0.1 255.255.255.255 192.168.0.2
ip route 10.0.0.2 255.255.255.255 192.168.0.3
ip route 192.168.1.0 255.255.255.0 Ethernet0/0
!
ip access-list extended access-list1
 permit ip any any
ip access-list extended access-list2
 permit ip any any dscp ef
!
```

Configuring Border-Router 1

```
dapr default
 border-router
  source-interface Loopback0
  route-manager 11.0.0.1
  authentication password 4 tnhtc92DXBhelxjYk8LWJrPV36S2i4ntXrpb4RFmfqY
!
interface Loopback0
 description BR-loopback
 ip address 10.0.0.1 255.255.255.255
!
interface Ethernet0/0
 description To-RM
 ip address 192.168.0.2 255.255.255.0
!
interface Ethernet1/0
 description To-Src-Host
 ip address 192.168.1.2 255.255.255.0
 dapr ingress
!
interface Serial2/0
 description WAN link
 ip address 192.168.10.2 255.255.255.0
 ip ospf cost 100
 serial restart-delay 0
 dapr egress link-group LG1
!
!
interface Serial3/0
 description WAN link
 ip address 192.168.11.2 255.255.255.0
 ip ospf cost 100
 dapr egress link-group LG1
!
router ospf 1
 network 10.0.0.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.255 area 0
 network 192.168.11.0 0.0.0.255 area 0
!
ip route 11.0.0.1 255.255.255.255 Ethernet0/0 192.168.0.1
```

Configuring Border-Router 2

```
dapr default
 border-router
  source-interface Loopback0
  route-manager 11.0.0.1
  authentication password 4 tnhtc92DXBhelxjYk8LWJrPV36S2i4ntXrpb4RFmfqY
!
interface Loopback0
 description BR-loopback
 ip address 10.0.0.2 255.255.255.255
!
interface Ethernet0/0
 description To-RM
 ip address 192.168.0.3 255.255.255.0
!
interface Ethernet1/0
 description To-Src-Host
```

```
 ip address 192.168.1.3 255.255.255.0
 dapr ingress
!
interface Serial2/0
 ip address 192.168.12.2 255.255.255.0
 ip ospf cost 100
 dapr egress link-group LG2
!
interface Serial3/0
 ip address 192.168.13.2 255.255.255.0
 ip ospf cost 100
 dapr egress link-group LG2
!
router ospf 1
 network 10.0.0.2 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.12.0 0.0.0.255 area 0
 network 192.168.13.0 0.0.0.255 area 0
!
ip route 11.0.0.1 255.255.255.255 Ethernet0/0 192.168.0.1
```

Show Commands for Route-Manager

```
Device#show dapr route-manager summary
Legend: BR - Border Router, RM - Route Manager
        U - Unmanaged, M - Managed, O - Out of policy, D - Marked for deletion
        R - Re-compute pending

  RM Status                       :  ACTIVE
  RM Address                      :  11.0.0.1
  BRs Registered/Configured       :  2/2
  Prefixes Learnt                 :  5
  Flow-groups Learnt (U/M/O/D/R)  :  4 (0/4/0/0/0)
  Thresholds (Min-BW, Max-Util)   :  500 kbps, 50%
  Flow-group Template             :  Source, Destination, DSCP

Device#show dapr route-manager border-router summary
Legend: S - Status
          D - Disconnected, C - Connected, R - Registered
        Nbr - Neighbor

-------------------------------------------------------------------------------
Address         S  Egress/  Nbr  Prefixes  Ingress App      Up-time
                   Ingress  BRs  Learnt    Flows   Routes
                   Intfs                  Learnt  Pushed
-------------------------------------------------------------------------------
10.0.0.1        R  2/1      1    3         2       2        8m 24s
10.0.0.2        R  2/1      1    3         2       2        8m 23s

Device#show dapr route-manager border-router neighbors
Legend: C - Connected, . - Disconnected
        1 - 10.0.0.2, 2 - 10.0.0.1

Inter BR Connectivity Matrix:
    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
1   C  C  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
2   C  C  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
3   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
4   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
5   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
6   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
7   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

```
8   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
9   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
10  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
11  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
12  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
13  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
14  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
15  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
16  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
17  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
18  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
19  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
20  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .


Device#show dapr route-manager border-router 10.0.0.1
Legend: BR - Border Router, BW - Bandwidth in kbps, SIdx - SNMP Ifindex

BR: 10.0.0.1
  Status                   :  REGISTERED
  Table Id                 :  0
  Egress/Ingress Intfs     :  2/1
  Neighbor BRs             :  1
  Prefixes Learnt          :  3
  Ingress Flows Learnt     :  2
  App/Flow-group Routes    :  2
  Up-time                  :  00:08:34
  Last FNF Template Rcvd   :  00:00:28
  Last RIB Update Rcvd     :  00:03:54
  FNF Export Seq Num       :  10
  FNF Export Pkts Missed   :  0
  Last Reset Reason        :  Reset RM

Ingress Interfaces:
  Interface-Name  SIdx  State
  Et1/0           5     UP

Egress Interfaces:
  Interface-Name  SIdx  State    BW(Cur/Avg) %Util(Cur/Avg) Link-Group
  Se2/0           9     UP        1544/1544     0/0         LG1
  Se3/0           13    UP        1544/1544     0/0         LG1

Neighbor BRs:
  Addresss        Tunnel      SIdx   State
  10.0.0.2        Tu0         19     UP


Device#show dapr route-manager border-router 10.0.0.2
Legend: BR - Border Router, BW - Bandwidth in kbps, SIdx - SNMP Ifindex

BR: 10.0.0.2
  Status                   :  REGISTERED
  Table Id                 :  0
  Egress/Ingress Intfs     :  2/1
  Neighbor BRs             :  1
  Prefixes Learnt          :  3
  Ingress Flows Learnt     :  2
  App/Flow-group Routes    :  2
  Up-time                  :  00:08:39
  Last FNF Template Rcvd   :  00:00:33
  Last RIB Update Rcvd     :  00:03:59
  FNF Export Seq Num       :  10
  FNF Export Pkts Missed   :  0
  Last Reset Reason        :  Reset RM
```

```
Ingress Interfaces:
  Interface-Name  SIdx  State
  Et1/0           5     UP

Egress Interfaces:
  Interface-Name  SIdx  State    BW(Cur/Avg) %Util(Cur/Avg) Link-Group
  Se2/0           9     UP        1544/1544     0/0         LG2
  Se3/0           13    UP        1544/1544     0/0         LG2


Device#show dapr route-manager link-groups
Legend: BR - Border Router

-----------------------------------------------
Link-group
   Members (BR, Egress Interface)
-----------------------------------------------
LG1
    10.0.0.1, Se2/0
    10.0.0.1, Se3/0
LG2
    10.0.0.2, Se2/0
    10.0.0.2, Se3/0


Device#show dapr route-manager route-table
Legend: BR - Border Router

---------------------------------------
Prefix
 BR  Next-Hop
---------------------------------------
12.0.0.0/16
  10.0.0.1 192.168.11.1, Se3/0
  10.0.0.2 192.168.12.1, Se2/0
  10.0.0.2 192.168.13.1, Se3/0
  10.0.0.1 192.168.10.1, Se2/0
192.168.10.0/24
  10.0.0.2 192.168.13.1, Se3/0
  10.0.0.2 192.168.12.1, Se2/0
192.168.11.0/24
  10.0.0.2 192.168.13.1, Se3/0
  10.0.0.2 192.168.12.1, Se2/0
192.168.12.0/24
  10.0.0.1 192.168.11.1, Se3/0
  10.0.0.1 192.168.10.1, Se2/0
192.168.13.0/24
  10.0.0.1 192.168.11.1, Se3/0
  10.0.0.1 192.168.10.1, Se2/0


Device#show dapr route-manager flow-groups
Legend: BR - Border Router, Rate - Flow rate(current) bps
        S - Status
          U - Unmanaged, M - Managed, O - Out of policy, D - Marked for deletion

Source           Destination     DSCP Rate    Up-time  S Egress-BR        Next-hop
13.0.0.1         12.0.0.1        def  0K      00:00:38 M 10.0.0.1         192.168.10.1,
Se2/0
13.0.0.1         12.0.0.2        def  0K      00:00:38 M 10.0.0.1         192.168.11.1,
Se3/0
13.0.0.1         12.0.0.3        def  0K      00:00:38 M 10.0.0.1         192.168.11.1,
Se3/0
```

```
13.0.0.1         12.0.0.4        def  0K      00:00:38 M 10.0.0.1         192.168.10.1,
Se2/0


Device#show dapr route-manager flow-groups detail
Legend: BR - Border Router, Rate - Flow rate(curr/avg) bps
        S - Flow State
          U - Unmanaged, M - Managed, O - Out of policy, D - Pending deletion
        Reason codes
         N - New flow-group, X - Expired, E - Invalid Egress
         I - Invalid Ingress, U - Path unreachable, NV - No viable path
         LO - Link out of policy, FO - Flow-group out of policy
         A - Admin deleted, IB - Ingress BR disconnected

--------------------------------------------------------------------------------
Flow-group(Source Destination DSCP):
 Attr:   IngressBR       Rate                           Up-time
 Curr: S EgressBR        Rate    Next-hop               Duration  Reason
 Prev: S EgressBR                Next-hop
--------------------------------------------------------------------------------
13.0.0.1, 12.0.0.1, def:
         10.0.0.1        0K/0K                          00:00:42
       M 10.0.0.1        0K      192.168.10.1, Se2/0    00:00:38  N
       U 10.0.0.1                -
13.0.0.1, 12.0.0.2, def:
         10.0.0.2        0K/0K                          00:00:42
       M 10.0.0.1        0K      192.168.11.1, Se3/0    00:00:38  N
       U 10.0.0.2                -
13.0.0.1, 12.0.0.3, def:
         10.0.0.1        0K/0K                          00:00:42
       M 10.0.0.1        0K      192.168.11.1, Se3/0    00:00:38  N
       U 10.0.0.1                -
13.0.0.1, 12.0.0.4, def:
         10.0.0.2        0K/0K                          00:00:42
       M 10.0.0.1        0K      192.168.10.1, Se2/0    00:00:38  N
       U 10.0.0.2
```

Show Commands for Border-Router

```
Device#show dapr border-router summary
Legend: BR - Border Router, RM - Route Manager

  BR Status                       : REGISTERED
  Local Address                   : 10.0.0.1
  RM Address                      : 11.0.0.1
  Egress Interfaces               : 2
  Ingress Interfaces              : 1
  Neighbor BRs                    : 1
  Last Successful Registration    : 00:15:10
  Last Stats Pull Request Rcvd    : 00:00:05
  Last RIB Pull Request Rcvd      : 00:00:35
  Last Flow Route Policy Rcvd     : 00:05:30
  Last Reset Reason               : Conn-Down
  Route-map Flows                 : 0
  Route-map Entries (Local/InterBR): 0 (0/0)
  Flow Record                     : dapr-flow-record
  Flow Exporter                   : dapr-flow-exporter
  Flow Monitor                    : dapr-flow-monitor
  Route Map                       : dapr-routemap


Device#show dapr border-router neighbors
```

```
Legend: SIdx - SNMP Ifindex

  Neighbor-BR      Tunnel       SIdx  Status
  10.0.0.2         Tunnel0      19    UP


Device#show dapr border-router interfaces
Legend: SIdx - SNMP Ifindex

Ingress Interfaces:
  Interface-Name   SIdx
  Et1/0            5

Egress Interfaces:
  Interface-Name   SIdx  Link-Group
  Se2/0            9     LG1
  Se3/0            13    LG1

Device#show dapr border-router interfaces metrics
Serial2/0
   Bandwidth kbps (Cur/Avg/Min/Max)  : 1544/1544/1544/1544
   % Utilization (Cur/Avg)           : 0/0
   Count (Pkt/Byte)                  : 0/0
Serial3/0
   Bandwidth kbps (Cur/Avg/Min/Max)  : 1544/1544/1544/1544
   % Utilization (Cur/Avg)           : 0/0
   Count (Pkt/Byte)                  : 0/0
Device#
```

Example for Configuring DAPR Co-located RM and BR

The following example show how to configure co-located RM and BR.

```
dapr default
 route-manager
  source-interface Loopback1
  authentication password 4 U28mHpS4suXM7r6q3U3E.oDXKCESijH3TSF6FHKrYHA
  link-thresholds
   max-utilization 50
   min-bandwidth 500
  border-routers
   10.0.0.2
 border-router
  source-interface Loopback0
  route-manager 10.0.0.100
  authentication password 4 U28mHpS4suXM7r6q3U3E.oDXKCESijH3TSF6FHKrYHA

interface Loopback0
 description BR-loopback
 ip address 10.0.0.2 255.255.255.255
end

interface Loopback1
 description RM-loopback
 ip address 10.0.0.100 255.255.255.255
end

Device#show dapr border-router summary
Legend: BR - Border Router, RM - Route Manager
  BR Status                        : REGISTERED
  Local Address                    : 10.0.0.2
```

```
  RM Address                       : 10.0.0.100
  RM Co-located                    : TRUE
```

Example for Configuring DAPR on RAR and PPPoE interfaces

DAPR is supported on RAR interfaces only in RAR bypass mode. Following is an example of RAR bypass mode configuration. For more information on RAR configuration, see the RAR Configuration Guide.

```
subscriber authorization enable
!
policy-map type service RAR-SERVICE1
 pppoe service manet_radio //pppoe service name must be manet_radio
```

Configure BBA Goup and Apply on the WAN Interface:

```
bba-groupGpppoe BBA-GROUP1
 virtual-template 1
 service profile RAR-SERVICE1
!
interface GigabitEthernet0/0/1
 ip address 22.23.23.1 255.255.0.0
 negotiation auto
 pppoe enable group BBA-GROUP1
```

Configure a Unique Loopback Interface for each Virtual-template:

```
interface Loopback1
 ip address 22.81.4.1 255.255.255.255
 ip ospf 100 area 0
 ip ospf cost 1000
```

Enable DAPR on the Virtual-template:

```
interface Virtual-Template1
 ip unnumbered Loopback1
 ip ospf 100 area 0
 ip ospf cost 1000
 no peer default ip address
 dapr egress link-group LG_1
```

Configure a VMI interface in Bypass Mode:

```
interface vmi1
 ip address 22.4.71.1 255.255.255.0
 physical-interface GigabitEthernet0/0/1
 mode bypass
```

Configure OSPF and Enable it on the Virtual-template:

```
router ospf 100
router-id 22.1.1.6
maximum-paths 20
```

Simulating RAR Radio Modem

RAR Radio modem can be simulated using a directly connected peer router. The following is an example of configuration required on the peer router to simulate an RAR Radio modem and the test commands to initiate a PPPoE session and change Radio bandwidth.

Note that the simulator only has RAR/PPPoE configuration and does not have any DAPR configuration.

```
subscriber authorization enable
!
```

```
policy-map type service RAR-SERVICE1
pppoe service manet_radio //pppoe service name must be manet_radio
```

Configure BBA Group and Apply on the WAN Interface:

```
bba-group pppoe BBA-GROUP1 virtual-template 1
service profile RAR-SERVICE1
!
interface GigabitEthernet0/0/3
ip address 22.39.39.1 255.255.0.0 negotiation auto
pppoe enable group BBA-GROUP1
```

Configure a Unique Loopback Interface for each Virtual-template:

```
interface Loopback1
ip address 22.81.7.3 255.255.255.255
ip ospf 100 area 0 ip ospf cost 1000
interface Virtual-Template1 ip unnumbered Loopback1
ip ospf 100 area 0 ip ospf cost 1000
no peer default ip address
```

Configure a VMI Interface in Bypass Mode:

```
interface vmi1
ip address 22.7.6.1 255.255.255.0
physical-interface GigabitEthernet0/0/3 mode bypass
```

Configure OSPF and Enabling it on the Virtual-template:

```
router ospf 100
router-id 22.1.1.7
```

Test Command on Simulator to Initiate a RAR/PPPoE Session

```
Simulator#test pppoe 1 1 g0/0/3
TEST: MAX: 1, CPS: 1
BRSR3#show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port                    VT  VA         State
           SID  LocMAC                                      VA-st      Type
    N/A      2  00fc.ba05.c273  Gi0/0/3                  1  Vi2.1      PTA
                00fc.ba3a.d3b1                              UP
```

Test Command on Simulator to Change RAR Link Bandwidth

```
Simulator#test pppoe session 2 padq mdr-scalar 1 max-data-rate 55 cdr-scalar 1 cur-data-rate
 55
```

Verifying the PPPoE Session

```
Device# show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port                    VT  VA         State
           SID  LocMAC                                      VA-st      Type
     46     37  00fc.ba05.c273  Gi0/0/1                  1  Vi1.1      PTA
                00fc.ba3a.d3b1                              UP


Device#show derived-config interface Vi1.1
Building configuration...
```

```
Derived configuration : 156 bytes
!
interface Virtual-Access1.1
ip unnumbered Loopback1
 ip ospf 100 area 0
 ip ospf cost 1000
 no peer default ip address
 dapr egress link-group LG_1
end

Device1#show int vi1.1
Virtual-Access1.1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of Loopback3 (22.81.7.3)
  MTU 1492 bytes, BW 100000 Kbit/sec, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open
  Open: IPCP
  PPPoE vaccess, cloned from Virtual-Template3
  Vaccess status 0x0
  Keepalive set (10 sec)
     89 packets input, 4706 bytes
     89 packets output, 4806 bytes
  Last clearing of "show interface" counters never
```

Debug Logs

Debug Logs for RM

The following are the debug logs for RM:

```
Device#debug dapr route-manager all
Device# debug dapr route-manager route-compute detail
debug dapr route-manager flow-collector detail

Device#show debugging
DAPR RM:
  DAPR RM Route-Compute debugging is on
  DAPR RM Route-Compute error debugging is on
  DAPR RM Route-Compute detail debugging is on
  DAPR RM Flow-Collector debugging is on
  DAPR RM Flow-Collector error debugging is on
  DAPR RM Flow-Collector detail debugging is on
  DAPR RM Events debugging is on
  DAPR RM Events error debugging is on
Device#

Device#configure terminal
DAPR-RM(config-dapr-instance)#route-manager
DAPR-RM(config-dapr-route-manager)#no shut
*Mar  6 11:09:14.174: %DAPR_RM-5-RM_STATUS: Active
Device#
```

Registration:

```
*Mar  6 11:09:36.445: DAPR-RM-EV: New BR connection, addr:10.0.0.1 port:45608
*Mar  6 11:09:36.445: %DAPR_RM-5-BR_STATUS: BR 10.0.0.1 CONNECTED
*Mar  6 11:09:36.445: DAPR-RM-EV: Received message from 10.0.0.1(fd:1)
*Mar  6 11:09:36.445: DAPR-RM-EV: Send message Registration Response to BR 10.0.0.1
```

```
*Mar  6 11:09:36.445: DAPR-RM-EV: Sent complete message to 10.0.0.1(fd:1)
Device#
*Mar  6 11:09:36.445: %DAPR_RM-5-BR_STATUS: BR 10.0.0.1 REGISTERED
DAPR-RM#
*Mar  6 11:09:37.446: DAPR-RM-EV: Received message from 10.0.0.1(fd:1)
*Mar  6 11:09:39.174: %DAPR_RM-6-BR_EVENT: BR Inter BR state event: 10.0.0.1
Device#
```

Periodic Information Pull:

```
*Mar  6 11:09:44.175: DAPR-RM-EV: Send message Pull Request to BR 10.0.0.1
*Mar  6 11:09:44.175: DAPR-RM-EV: Sent complete message to 10.0.0.1(fd:1)
*Mar  6 11:09:44.175: DAPR-RM-EV: Received message from 10.0.0.1(fd:1)

*Mar  6 11:10:14.174: DAPR-RM-EV: Send message Pull Request to BR 10.0.0.1
*Mar  6 11:10:14.174: DAPR-RM-EV: Sent complete message to 10.0.0.1(fd:1)
*Mar  6 11:10:14.174: DAPR-RM-EV: Received message from 10.0.0.1(fd:1)
```

Route-compute for Discovered Flow-group:

```
*Mar  6 11:10:49.175:   Viable paths:
*Mar  6 11:10:49.175:           Path:{10.0.0.1, [0]192.168.10.1, 9}, Pref:1, BW:1544,
Hr:1544, Util:0, TCC: 0
*Mar  6 11:10:49.175:           Path:{10.0.0.1, [0]192.168.11.1, 13}, Pref:1, BW:1544,
Hr:1544, Util:0, TCC: 0
*Mar  6 11:10:49.175: %DAPR_RM-6-APP_RT_INSTALL: TC[P]:{192.168.1.1/32, 12.0.0.1/32, default}
 on 10.0.0.1[0] (BW:0) Path:{10.0.0.1, [0]192.168.10.1, 9}
*Mar  6 11:10:49.175: DAPR-RM-EV: Send message FG Route Push to BR 10.0.0.1
*Mar  6 11:10:49.175: DAPR-RM-EV: Sent complete message to 10.0.0.1(fd:1)
```

Route-delete on Flow Expiry:

```
*Mar  6 11:12:16.922: DAPR-RM-FC-DETAIL: delete flow - reason 2
*Mar  6 11:12:19.176: %DAPR_RM-6-APP_RT_DEL: FG[D]:{192.168.1.1, 12.0.0.1, default} on
10.0.0.1 (BW:0)
*Mar  6 11:12:19.176: DAPR-RM-EV: Send message FG Route Push to BR 10.0.0.1
*Mar  6 11:12:19.176: DAPR-RM-EV: Sent complete message to 10.0.0.1(fd:1)
```

Debug Logs for BR

The following are the debug logs for BR:

```
Device#show debugging
Device:
DAPR BR All debugging is on DAPR BR Events debugging is on
DAPR BR Events Error debugging is on DAPR BR Flow-Route debugging is on
DAPR BR Flow-Route Error debugging is on DAPR BR RIB debugging is on
DAPR BR RIB Error debugging is on DAPR BR Flow-Export debugging is on
DAPR BR Flow-Export Error debugging is on DAPR BR Inter-BR Tunnel debugging is on
DAPR BR Inter-BR Tunnel Error debugging is on DAPR BR WAN-Metric debugging is on
DAPR BR WAN-Metric Error debugging is on
```

BR Shutdown:

```
Device#conf t
Device (config)#dapr default
Device(config-dapr-instance)#border-router Device(config-dapr-border-router)#shudown

*Mar 6 11:08:03.003: %DAPR_BR-5-STATUS: shutdown
*Mar 6 11:08:03: DAPR-BR-EV: Handle config shutdown notification
*Mar 6 11:08:03: DAPR-BR-EV: Enqueue Connection Close Request
*Mar 6 11:08:03: DAPR-BR-EV: Handle BR-RM event for disconnect
*Mar 6 11:08:03: DAPR-BR-EV: Received BR-RM Connection Close, reason: Config shutdown
```

```
*Mar 6 11:08:03: DAPR-BR-EV: Cleanup BR info
*Mar 6 11:08:03: DAPR-BR-EV: BR-RM Connection Closed by BR DAPR-BR1#
```

TCP Control Connection to RM:

```
Device#configure terminal
Device(config)#dapr default
Device(config-dapr-instance)#border-router
Device(config-dapr-border-router)#no shudown

*Mar 6 11:09:36: DAPR-BR-EV: Handle config criteria met notification
*Mar 6 11:09:36: DAPR-BR-EV: Enqueue Connection Request
*Mar 6 11:09:36: DAPR-BR-FR: Handle config criteria met Notification
*Mar 6 11:09:36: DAPR-BR-EV: Handle BR-RM event for connect
*Mar 6 11:09:36: DAPR-BR-EV: Received BR-RM Connection Request
*Mar 6 11:09:36: DAPR-BR-RIB: Check RM route validity
*Mar 6 11:09:36: DAPR-BR-RIB: lookup returned out_idb:Ethernet0/0 for tableid:0
rm_addr:11.0.0.1
*Mar 6 11:09:36: DAPR-BR-RIB: rm route is via Ethernet0/0
*Mar 6 11:09:36: DAPR-BR-RIB: Route to RM is VALID
*Mar 6 11:09:36: DAPR-BR-EV: Connect to RM, local: 10.0.0.1(0), remote: 11.0.0.1(17749),
idb:Loopback0
*Mar 6 11:09:36: DAPR-BR-EV: Set tableid 0
*Mar 6 11:09:36: DAPR-BR-EV: socket 0 connect status: -1 errno: 11
*Mar 6 11:09:36: DAPR-BR-EV: Connect to RM PENDING on fd 0
*Mar 6 11:09:36: DAPR-BR-EV: BR-RM Connection IN PROGRESS
*Mar 6 11:09:36: DAPR-BR-EV: Handle BR-RM Connection Pending Request
*Mar 6 11:09:36: DAPR-BR-EV: BR-RM(11.0.0.1) channel progress->connected, make connection
UP
*Mar 6 11:09:36: DAPR-BR-EV: BR-RM Connection SUCCESSFUL
*Mar 6 11:09:36.445: %DAPR_BR-5-STATUS: CONNECTED
*Mar 6 11:09:36: DAPR-BR-FR: Handle connection UP
```

Registration:

```
*Mar 6 11:09:36: DAPR-BR-EV: Send message Registration Request to RM 11.0.0.1(fd:0)
*Mar 6 11:09:36: DAPR-BR-EV: Sent complete message to 11.0.0.1(fd:0)
*Mar 6 11:09:36: DAPR-BR-EV: Registration request sent to RM
*Mar 6 11:09:36: DAPR-BR-EV: Received message from 11.0.0.1(fd:0)
*Mar 6 11:09:36: DAPR-BR-EV: Received msg Registration Response from RM
*Mar 6 11:09:36.445: %DAPR_BR-5-STATUS: REGISTERED
```

Inter-BR Tunnel Creation:

```
*Mar 6 11:09:36: DAPR-BR-RIB: Check Inter-BR route validity for 10.0.0.2
*Mar 6 11:09:36: DAPR-BR-RIB: lookup returned out_idb:Ethernet1/0 for tableid:0
br_addr:10.0.0.2
*Mar 6 11:09:36: DAPR-BR-RIB: inter-br route is via Ethernet1/0
*Mar 6 11:09:36: DAPR-BR-INTER-BR: Tunnel ceate to 10.0.0.2: Succefully created inter BR
tunnel Tunnel0
Enabling egress Netflowv9 on DAPR egress interfaces:
Mar 6 11:09:36: DAPR-BR-FLOW-EXP: Created Flow record dapr-flow-record
*Mar 6 11:09:36: DAPR-BR-FLOW-EXP-ERR: Flow exporter create: Exporter mtu 16384
*Mar 6 11:09:36: DAPR-BR-FLOW-EXP: Created DAPR owned fnf exporter dapr-flow-exporter
(11.0.0.1:9995)
*Mar 6 11:09:36: DAPR-BR-FLOW-EXP: Flow monitor create sucess: Monitor name dapr-flow-monitor
*Mar 6 11:09:36: DAPR-BR-FLOW-EXP: Attached monitor dapr-flow-monitor on interface Serial2/0:
*Mar 6 11:09:36: DAPR-BR-FLOW-EXP: Attached monitor dapr-flow-monitor on interface Serial3/0:
```

Start Monitoring DAPR Egress Interfaces:

```
Mar 6 11:09:44: DAPR-BR-EV: Received message from 11.0.0.1(fd:0)
*Mar 6 11:09:44: DAPR-BR-EV: Received msg Pull Request from RM
*Mar 6 11:09:44: DAPR-BR-RIB: RIB walk and populate
*Mar 6 11:09:44: DAPR-BR-RIB: Total prefixes:3 max:1000
*Mar 6 11:09:44: DAPR-BR-RIB: RIB walk and populate SUCCESS, prefixes 3 routes 6
```

```
*Mar 6 11:09:44: DAPR-BR-EV: Send message Pull Response to RM 11.0.0.1(fd:0)
*Mar 6 11:09:44: DAPR-BR-EV: Sent complete message to 11.0.0.1(fd:0)
```

Periodic Information Pull Request from RM:

```
Mar 6 11:09:44: DAPR-BR-EV: Received message from 11.0.0.1(fd:0)
*Mar 6 11:09:44: DAPR-BR-EV: Received msg Pull Request from RM
*Mar 6 11:09:44: DAPR-BR-RIB: RIB walk and populate
*Mar 6 11:09:44: DAPR-BR-RIB: Total prefixes:3 max:1000
*Mar 6 11:09:44: DAPR-BR-RIB: RIB walk and populate SUCCESS, prefixes 3 routes 6
*Mar 6 11:09:44: DAPR-BR-EV: Send message Pull Response to RM 11.0.0.1(fd:0)
*Mar 6 11:09:44: DAPR-BR-EV: Sent complete message to 11.0.0.1(fd:0)
```

Periodic Sampling of DAPR Egress Bandwith and Utilization:

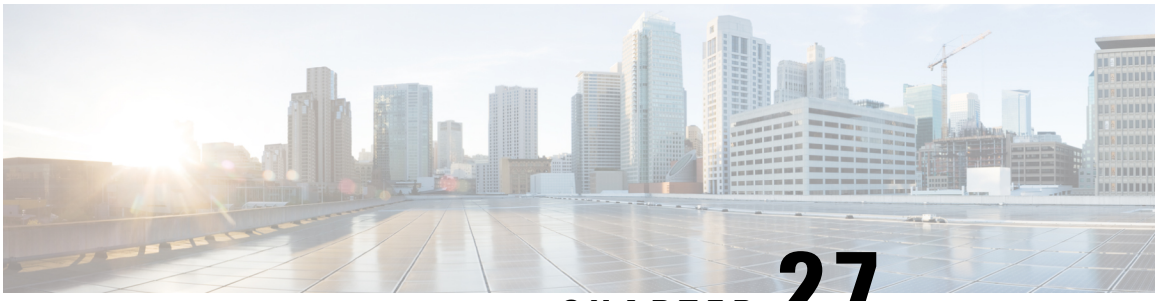
```
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Current Sample: (max samples = 3, curr_idx = 0,
next_idx = 1)
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Current sample utilization 0 (index 0)
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Utilization Samples Collected:
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Average Utilization of collected samples: 0
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Current Sample: (max samples = 3, curr_idx = 0,
next_idx = 1)
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Current sample utilization 0 (index 0)
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Utilization Samples Collected:
*Mar 6 11:09:46: DAPR-BR-WAN-METRIC: Average Utilization of collected samples: 0
```

Periodic Information Pull Request from RM:

```
Periodic information pull request from RM:
*Mar 6 11:10:14: DAPR-BR-EV: Received message from 11.0.0.1(fd:0)
*Mar 6 11:10:14: DAPR-BR-EV: Received msg Pull Request from RM
*Mar 6 11:10:14: DAPR-BR-EV: Send message Pull Response to RM 11.0.0.1(fd:0)
*Mar 6 11:10:14: DAPR-BR-EV: Sent complete message to 11.0.0.1(fd:0)
```

Route Push Message from RM to BR:

```
Mar 6 11:14:19: DAPR-BR-EV: Received message from 11.0.0.1(fd:0)
*Mar 6 11:14:19: DAPR-BR-EV: Received msg FG Route Push from RM
*Mar 6 11:14:19: DAPR-BR-FR: ***BEGIN***
*Mar 6 11:14:19: DAPR-BR-FR: Remove route map entries, total: 1
*Mar 6 11:14:19: DAPR-BR-FR: No new entries received
*Mar 6 11:14:19: DAPR-BR-FR: calling rmap batch commit
*Mar 6 11:14:19: DAPR-BR-FR: ***END:SUCCESS***
Device#
```



CHAPTER 27

Unicast Reverse Path Forwarding Strict Mode

The Unicast Reverse Path Forwarding feature limits the malicious traffic on a network. This feature enables devices to verify the reachability of the source address in packets that are being forwarded and limit the appearance of spoofed or malformed addresses on a network. If the source IP address is not valid, Unicast Reverse Path Forwarding (RPF) discards the packet.

This module describes the Unicast Reverse Path Forwarding feature.

Prerequisites for Unicast Reverse Path Forwarding

- Unicast Reverse Path Forwarding (RPF) requires Cisco Express Forwarding to function properly on a device.
- Prior to configuring Unicast RPF, you must configure the following access control lists (ACLs):
  - Configure standard or extended ACL to mitigate the transmission of invalid IP addresses (by performing egress filtering). Configuring standard or extended ACLs permit only valid source addresses to leave your network and enter the Internet.
  - Configure standard or extended ACL entries to drop (deny) packets that have invalid source IP addresses (by performing ingress filtering). Invalid source IP addresses include the following types:
    - Broadcast addresses (including multicast addresses)
    - Loopback addresses
    - Private addresses (RFC 1918, *Address Allocation for Private Internets*)
    - Reserved addresses
    - Source addresses that fall outside the range of valid addresses that are associated with the protected network

Restrictions for Unicast Reverse Path Forwarding

- Unicast RPF does not support access control list (ACL) templates.

The following basic restrictions apply to multihomed clients:

- Clients should not be multihomed on the same device because multihoming defeats the purpose of creating a redundant service for a client.
- Ensure that packets that flow up the link (out to the Internet) match the route advertised out of the link. Otherwise, Unicast RPF filters these packets as malformed packets.

Information About Unicast Reverse Path Forwarding

Overview of Unicast Reverse Path Forwarding

The Unicast Reverse Path Forwarding feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack verifiable IP source addresses. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter these attacks. For ISPs that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table, thereby protecting the network of the ISP, ISP customers, and the Internet.

Unicast RPF Operation

When Unicast RPF is enabled on an interface of a device, the device examines all packets received as input on that interface to ensure that the source address and source interface information appears in the routing table and matches the interface on which packets are received. This ability to “look backwards” is available only when Cisco Express Forwarding is enabled on a device because the lookup relies on the presence of a Forwarding Information Base (FIB). Cisco Express Forwarding generates a FIB as part of its operation.

Note Unicast RPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

Unicast RPF does a reverse lookup in the Cisco Express Forwarding table to check if any packet received at the interface of a device arrives on the best return path (or return route) to the source of the packet. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. No reverse path route on the interface from which the packet was received can mean that the source address was modified.

Note With Unicast RPF, all equal-cost "best" return paths are considered valid. Unicast RPF supports multiple return paths, provided that each path is equal to the others in terms of the routing cost (such as number of hops, weights, and so on) and the route is available in the FIB. Unicast RPF also functions where Enhanced Interior Gateway Routing Protocol (EIGRP) variants are used.

Before forwarding a packet that is received at the interface on which Unicast RPF and ACLs have been configured, Unicast RPF does the following checks:

1. If input ACLs are configured on the inbound interface.
2. If the packet has arrived on the best return path to the source by doing a reverse lookup in the FIB table.
3. Does a lookup of the Cisco Express Forwarding table for packet forwarding.
4. Checks output ACLs on the outbound interface.
5. Forwards the packet.

Access Control Lists and Logging

When you configure an access control list (ACL) and a packet fails the Unicast RPF check, the Unicast RPF checks the ACL to see if the packet should be dropped (by using a deny statement in the ACL) or forwarded (by using a permit statement in the ACL). Regardless of whether the packet is dropped or forwarded, the packet is counted in the global IP traffic statistics for Unicast RPF drops and in the interface statistics for Unicast RPF.

If no ACL is configured, the device drops the forged or malformed packet immediately, and no ACL logging occurs. The device and the interface Unicast RPF logging counters are updated.

To log Unicast RPF events, specify the logging option for ACL entries. Using the log information, administrators can view source addresses that are used in an attack, the time at which packets arrived at an interface, and so on.

Caution Logging requires CPU and memory resources. Logging Unicast RPF events for attacks that have a high rate of forged packets can degrade the performance of a device.

Per-Interface Statistics

Each time a packet is dropped or forwarded at an interface, that information is counted two ways: globally on the router and at each interface where you have applied Unicast RPF. Global statistics on dropped packets provide information about potential attacks on the network; however, these global statistics do not help to specify which interface is the source of the attack.

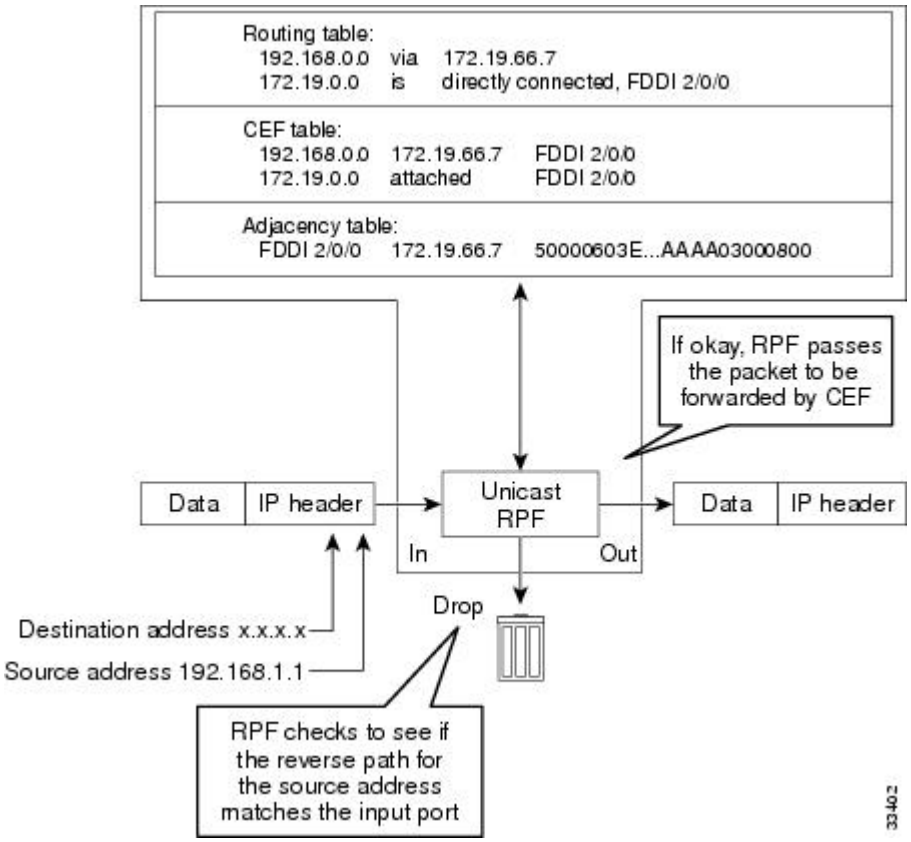
Per-interface statistics allow network administrators to track two types of information about malformed packets: Unicast RPF drops and Unicast RPF suppressed drops. Statistics on the number of packets that Unicast RPF drops help to identify the interface that is the entry point of the attack. The Unicast RPF drop count tracks the number of drops at the interface. The Unicast RPF suppressed drop count tracks the number of packets that failed the Unicast RPF check but were forwarded because of the permit permission set up in the ACL.

Using the drop count and suppressed drop count statistics, a network administrator can takes steps to isolate the attack at a specific interface.

Note Judicious use of ACL logging can further identify the address or addresses that are being dropped by Unicast RPF.

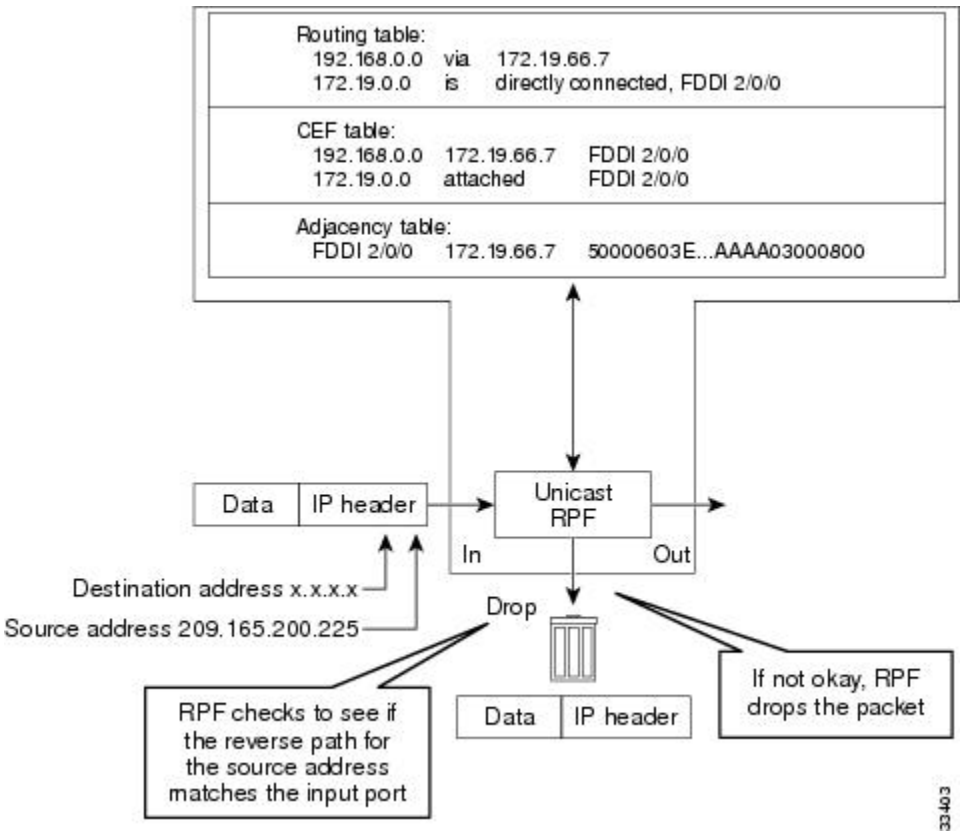
The figure below illustrates how Unicast RPF and CEF work together to validate IP source addresses by verifying packet return paths. In this example, a customer has sent a packet having a source address of 192.168.1.1 from interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 192.168.1.1 has a path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. If there is no matching path, the packet is dropped.

Figure 25: Unicast RPF Validating IP Source Addresses



The figure below illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. In this case, there is no reverse entry in the routing table that routes the customer packet back to source address 209.165.200.225 on interface FDDI 2/0/0, and so the packet is dropped.

Figure 26: Unicast RPF Dropping Packets That Fail Verification



Rules for Implementing Unicast RPF

The following rules apply when implementing Unicast Reverse Path Forwarding (RPF):

- Packets must be received at an interface that has the best return path (route) to the packets' source. This process is called symmetric routing. A route in the Forwarding Information Base (FIB) must match the route to the receiving interface. Add a route in the FIB through dynamic or static routing or by using a network statement.
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and can be applied at the input interface of a device at the upstream end of a connection.

Network administrators can use Unicast RPF for their customers and also for their downstream network or ISP, even if the downstream network or ISP has other connections to the Internet.

Caution Using optional Border Gateway Protocol (BGP) attributes, such as weight and local preference, the best path back to source addresses can be modified. The best path modification will affect the operation of Unicast RPF.

The following sections provides information about the implementation of Unicast RPF:

Security Policy and Unicast RPF

When determining how to deploy Unicast Reverse Path Forwarding (RPF), consider the following points:

- Apply Unicast RPF at the downstream interface, away from the larger portion of the network, preferably at the edges of your network. The further you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying sources of spoofed addresses. For example, applying Unicast RPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but Unicast RPF does not help in identifying the source of the attack. Applying Unicast RPF at the network access server helps to limit the scope of the attack and trace the source of the attack. However, deploying Unicast RPF across many sites adds to the administration cost of operating a network.
- When you deploy Unicast RPF on many entities on a network (for example, across the Internet, intranet, and extranet resources), you have better chances of mitigating large-scale network disruptions throughout the Internet community, and of tracing the source of an attack.
- Unicast RPF does not inspect IP packets that are encapsulated in tunnels, such as the generic routing encapsulation (GRE), Layer 2 Tunneling Protocol (L2TP), or Point-to-Point Tunneling Protocol (PPTP). Configure Unicast RPF on a home gateway so that Unicast RPF processes network traffic only after tunneling and encryption layers are stripped off from the packets.

Ingress and Egress Filtering Policy for Unicast RPF

Unicast Reverse Path Forwarding (RPF) can be more effective at mitigating spoofing attacks when combined with a policy of ingress and egress filtering by using access control lists (ACLs).

Ingress filtering applies filters to traffic that is received at a network interface from either internal or external networks. With ingress filtering, packets that arrive from other networks or the Internet and that have a source address that matches a local network or private or broadcast addresses are dropped. For example, in ISP environments, ingress filtering can be applied to traffic that is received at a device from either a client (customer) or the Internet.

Egress filtering applies filters to the traffic that exits a network interface (the sending interface). By filtering packets on devices that connect your network to the Internet or to other networks, you can permit only packets with valid source IP addresses to leave your network.

For more information on network filtering, refer to RFC 2267, *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*.

Where to Use Unicast RPF

Unicast Reverse Path Forwarding (RPF) can be used in any “single-homed” environment where there is essentially only one access point out of the network, which means that there is only one upstream connection to the network. Networks having one access point offer the best example of symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet. Unicast RPF is best used at the network perimeter for Internet, intranet, or extranet environments, or in ISP environments for customer network terminations.

The following sections describe two sample network environments in which Unicast RPF is implemented:

Enterprise Networks with a Single Connection to an ISP

In enterprise networks, you can use Unicast Reverse Path Forwarding (RPF) to filter traffic at the input interface (a process called ingress filtering) to protect from malformed packets that arrive from the Internet.

Traditionally, local networks that have one connection to the Internet use access control lists (ACLs) at the receiving interface to prevent spoofed packets from entering their local network.

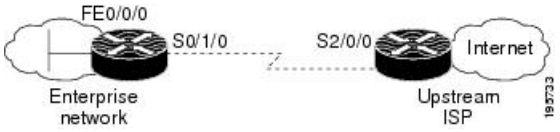
ACLs work well for single-homed customers. However, when ACLs are used as ingress filters, the following two commonly referenced limitations apply:

- Packet-per-second (PPS) performance at very high packet rates
- ACL maintenance (whenever there are new addresses added to the network)

Unicast RPF addresses both the limitations described above. With Unicast RPF, ingress filtering is done at Cisco Express Forwarding PPS rates. Because Unicast RPF uses the Forwarding Information Base (FIB), ACL maintenance is not required, and thus, the administration overhead of traditional ACLs is reduced. The following figure and example demonstrate how Unicast RPF is configured for ingress filtering.

The figure below illustrates an enterprise network that has a single link to an upstream ISP. In this example, Unicast RPF is applied at GigabitEthernet interface 1/0/2 on the enterprise device for protection from malformed packets arriving from the Internet. Unicast RPF is also applied at GigabitEthernet interface 1/0/2 on the ISP device for protection from malformed packets arriving from the enterprise network.

Figure 27: Enterprise Network Using Unicast RPF for Ingress Filtering



A typical configuration on an ISP device that uses the topography in the figure above would be as follows:

```
ip cef
interface loopback 0
  description Loopback interface on Gateway Device 2
  ip address 192.168.3.1 255.255.255.255
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
interface GigabitEthernet 1/0/2
  description 128K HDLC link to ExampleCorp WT50314E  R5-0
  bandwidth 128
  ip unnumbered loopback 0


  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
!
ip route 192.168.10.0 255.255.252.0 GigabitEthernet 1/0/2
```

The gateway device configuration of the enterprise network will be similar to the following:

```
ip cef
interface FastEthernet 0/0/0
 description ExampleCorp LAN
 ip address 192.168.10.1 255.255.252.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
!
interface GigabitEthernet 1/0/2
```

```
 description 128K HDLC link to ExampleCorp Internet Inc WT50314E  C0
 bandwidth 128
 ip unnumbered FastEthernet 0/0/0


 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet 1/0/2
```

Notice that Unicast RPF works with a single default route. There are no additional routes or routing protocols. Network 192.168.10.0/22 is a connected network. Hence, packets coming from the Internet with a source address in the network 192.168.10.0/22 will be dropped by Unicast RPF.

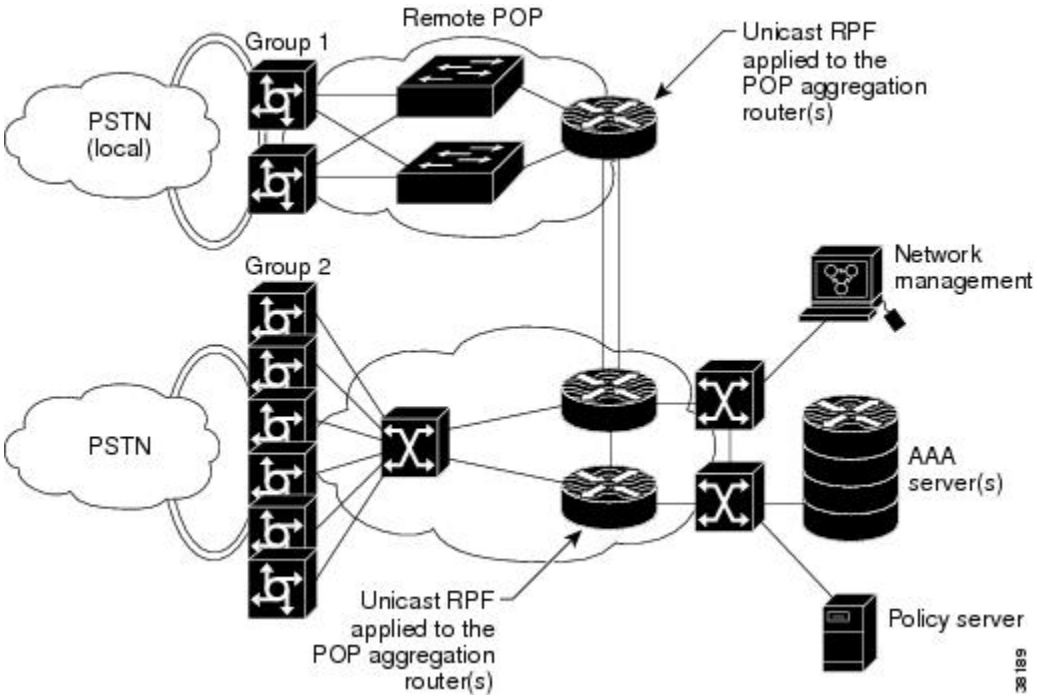
Applying Unicast RPF to Network Access Servers

If a network access server supports Cisco Express Forwarding, Unicast RPF will work on that network. A network access server (NAS) allows users to access a network by checking the credentials of the users accessing the network. Aggregation devices support Unicast RPF with single-homed clients. Unicast RPF works well on leased lines or on a digital subscriber line (DSL), ISDN, or public switched telephone network (PSTN) customer connections that are connected to the Internet. Dialup connections are a big source of denial of service (Dos) attacks that use forged IP addresses.

Aggregation devices need routing prefixes information (IP address block) for routing traffic. In the topology described below, aggregation devices do not have a full Internet routing table, and as a result, Unicast RPF uses the information configured or redistributed by the Interior Gateway Protocol (IGP) or Internal Border Gateway Protocol (IBGP) (based on how customer routes are added to the network) to route traffic. Unicast RPF is applied upstream on the customer dialup connection device that is on the receiving (input) interfaces of ISP aggregation devices.

The figure below illustrates how Unicast RPF is applied to aggregation and access devices for an ISP or point of presence (PoP) with ISP devices providing dialup connections.

Figure 28: Unicast RPF Applied to PSTN/ISDN Customer Connections



Routing Table Requirements

Unicast Reverse Path Forwarding (RPF) uses the routing information in Cisco Express Forwarding tables for routing traffic. The amount of routing information that must be available in Cisco Express Forwarding tables depends on the device where Unicast RPF is configured and the functions the device performs in the network. For example, in an ISP environment where a device is a leased-line aggregation device for customers, the information about static routes that are redistributed into the Interior Gateway Protocol (IGP) or Internal Border Gateway Protocol (IBGP) (depending on which technique is used in the network) is required in the routing table. Because Unicast RPF is configured on customer interfaces, only minimal routing information is required. If a single-homed ISP configures Unicast RPF on the gateway to the Internet, the full Internet routing table information is required by Unicast RPF to help protect the ISP from external denial of service (DoS) attacks that use addresses that are not in the Internet routing table.

Where Not to Use Unicast RPF

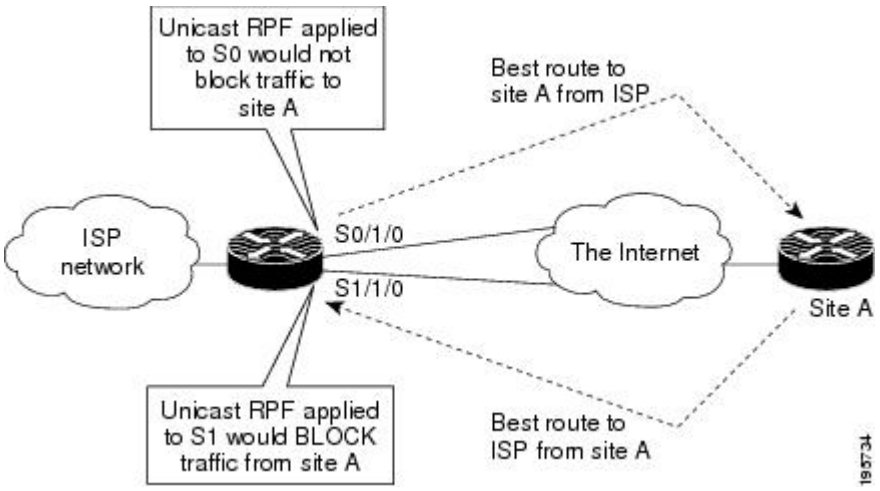
Do not use Unicast Reverse Path Forwarding (RPF) on interfaces that are internal to a network. Internal interfaces are likely to have routing asymmetry (see the figure below), which means that there can be multiple routes to the source of a packet. Unicast RPF is applied only where there is a natural or configured symmetry.

For example, devices at the edge of an ISP network are more likely to have symmetrical reverse paths than devices that are in the core of an ISP network. The best forwarding path to forward packets from devices that are at the core of an ISP network may not be the best forwarding path that is selected for packets that are returned to the device.

We recommend that you do not apply Unicast RPF where there is a chance of asymmetric routing, unless you configure access control lists (ACLs) to allow the device to accept incoming packets. ACLs permit the use of Unicast RPF when packets arrive through specific, less-optimal asymmetric input paths.

The figure below illustrates how Unicast RPF can block legitimate traffic in an asymmetric routing environment.

Figure 29: Unicast RPF Blocking Legitimate Traffic in an Asymmetric Routing Environment



Unicast RPF with BOOTP and DHCP

Unicast RPF allows packets with 0.0.0.0 as the source IP address and 255.255.255.255 as the destination IP address to pass through a network to enable Bootstrap Protocol (BOOTP) and DHCP functions to work properly when Unicast RPF is configured.

How to Configure Unicast Reverse Path Forwarding

Configuring Unicast RPF

Before you begin

To use Unicast Reverse Path Forwarding, you must configure a device for Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching. If Cisco Express Forwarding is not enabled globally on a device, Unicast RPF will not work on that device. If Cisco Express Forwarding is running on a device, individual interfaces on the device can be configured with other switching modes. Unicast RPF is an input-side function that is enabled on an interface or subinterface that supports any type of encapsulation, and Unicast RPF operates on IP packets that are received by the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **interface** *slot/subslot/port*
5. **exit**
6. **end**
7. **show cef interface** [*type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Device(config)# ip cef distributed`	Enables Cisco Express Forwarding or distributed Cisco Express Forwarding on a device.
Step 4	**interface** *slot*/*subslot*/*port* **Example:** `Device(config)# interface GigabitEthernet 0/0`	Selects the input interface on which you want to apply Unicast Reverse Path Forwarding and enters interface configuration mode. • The interface that is configured is the receiving interface, which allows Unicast RPF to verify the best return path before forwarding a packet to the next destination.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.
Step 7	**show cef interface** [*type number*] **Example:** `Device# show cef interface`	Displays detailed Cisco Express Forwarding information for a specified interface or for all interfaces.

Example:

Troubleshooting Tips

HSRP Failure

The failure to disable Unicast RPF before disabling Cisco Express Forwarding can cause a Hot Standby Router Protocol (HSRP) failure. If you want to disable Cisco Express Forwarding on a device, you must first disable Unicast RPF.

Configuration Examples for Unicast Reverse Path Forwarding

Example: Configuring Unicast RPF

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Unicast RPF command descriptions	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Cisco Express Forwarding commands	Cisco IOS IP Switching Command Reference

Standards & RFCs

Standard/RFC	Title
RFC 1918	*Address Allocation for Private Internets*
RFC 2267	*Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

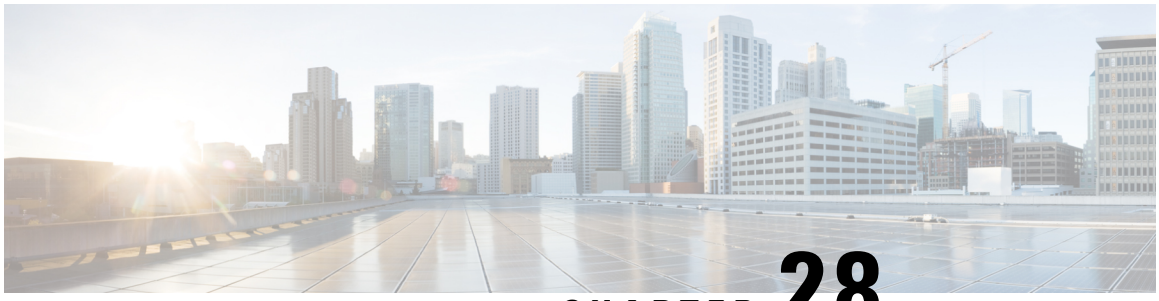
Feature Information for Unicast Reverse Path Forwarding

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 35: Feature Information for Unicast Reverse Path Forwarding

Feature Name	Releases	Feature Information
Unicast Reverse Path Forwarding	Cisco IOS XE Release 2.1	The Unicast Reverse Path Forwarding feature limits the malicious traffic on a network. This feature enables devices to verify the reachability of the source address in packets that are being forwarded and limit the appearance of spoofed or malformed addresses on a network. If the source IP address is not valid, Unicast Reverse Path Forwarding (RPF) discards the packet.



CHAPTER 28

Unicast Reverse Path Forwarding ACL Support

The Unicast Reverse Path Forwarding feature helps to mitigate problems that are caused by malformed or forged IP source addresses that pass through a device. The Unicast Reverse Path Forwarding ACL Support feature adds the access control list (ACL) support to the Unicast Reverse Path Forwarding feature. With the ACL support, Unicast Reverse Path Forwarding (RPF) can determine whether to drop or to forward data packets that have malformed or forged IP source addresses.

This module describes the ACL support for Unicast RPF.

Prerequisites for Unicast Reverse Path Forwarding ACL Support

- Unicast RPF requires Cisco Express Forwarding to function properly on a device.
- Prior to configuring Unicast RPF, you must configure the following ACLs:
  - Configure standard or extended ACLs to mitigate the transmission of invalid IP addresses (by performing egress filtering). Configuring standard or extended ACLS, permit only valid source addresses to leave your network and enter the Internet.
  - Configure standard or extended ACL entries to drop (deny) packets that have invalid source IP addresses (by performing ingress filtering). Invalid source IP addresses include the following types:
    - Broadcast addresses (including multicast addresses)
    - Loopback addresses
    - Private addresses (RFC 1918, *Address Allocation for Private Internets*)
    - Reserved addresses
    - Source addresses that fall outside the range of valid addresses associated with a protected network

- Configure standard or extended ACL entries to forward (permit) packets that fail the Unicast RPF checks and allow specific traffic from known asymmetric routed sources.

- Configure ACLs to track Unicast RPF events to provide additional information about network attacks.

Restrictions for Unicast Reverse Path Forwarding ACL Support

ACL templates are not supported.

Information About Unicast Reverse Path Forwarding ACL Support

Unicast RPF Operation

When Unicast RPF is enabled on an interface of a device, the device examines all packets received as input on that interface to ensure that the source address and source interface information appears in the routing table and matches the interface on which packets are received. This ability to "look backwards" is available only when Cisco Express Forwarding is enabled on a device because the lookup relies on the presence of a Forwarding Information Base (FIB). Cisco Express Forwarding generates a FIB as part of its operation.

Note Unicast RPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

Unicast RPF does a reverse lookup in the Cisco Express Forwarding table to check if any packet received at the interface of a device arrives on the best return path (or return route) to the source of the packet. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. No reverse path route on the interface from which the packet was received can mean that the source address was modified.

Note With Unicast RPF, all equal-cost "best" return paths are considered valid. Unicast RPF supports multiple return paths, provided that each path is equal to the others in terms of the routing cost (such as number of hops, weights, and so on) and the route is available in the FIB. Unicast RPF also functions where Enhanced Interior Gateway Routing Protocol (EIGRP) variants are used.

Before forwarding a packet that is received at the interface on which Unicast RPF and ACLs have been configured, Unicast RPF does the following checks:

1. If input ACLs are configured on the inbound interface.
2. If the packet has arrived on the best return path to the source by doing a reverse lookup in the FIB table.
3. Does a lookup of the Cisco Express Forwarding table for packet forwarding.
4. Checks output ACLs on the outbound interface.
5. Forwards the packet.

Access Control Lists and Logging

When you configure an access control list (ACL) and a packet fails the Unicast RPF check, the Unicast RPF checks the ACL to see if the packet should be dropped (by using a deny statement in the ACL) or forwarded (by using a permit statement in the ACL). Regardless of whether the packet is dropped or forwarded, the packet is counted in the global IP traffic statistics for Unicast RPF drops and in the interface statistics for Unicast RPF.

If no ACL is configured, the device drops the forged or malformed packet immediately, and no ACL logging occurs. The device and the interface Unicast RPF logging counters are updated.

To log Unicast RPF events, specify the logging option for ACL entries. Using the log information, administrators can view source addresses that are used in an attack, the time at which packets arrived at an interface, and so on.

Caution Logging requires CPU and memory resources. Logging Unicast RPF events for attacks that have a high rate of forged packets can degrade the performance of a device.

Per-Interface Statistics

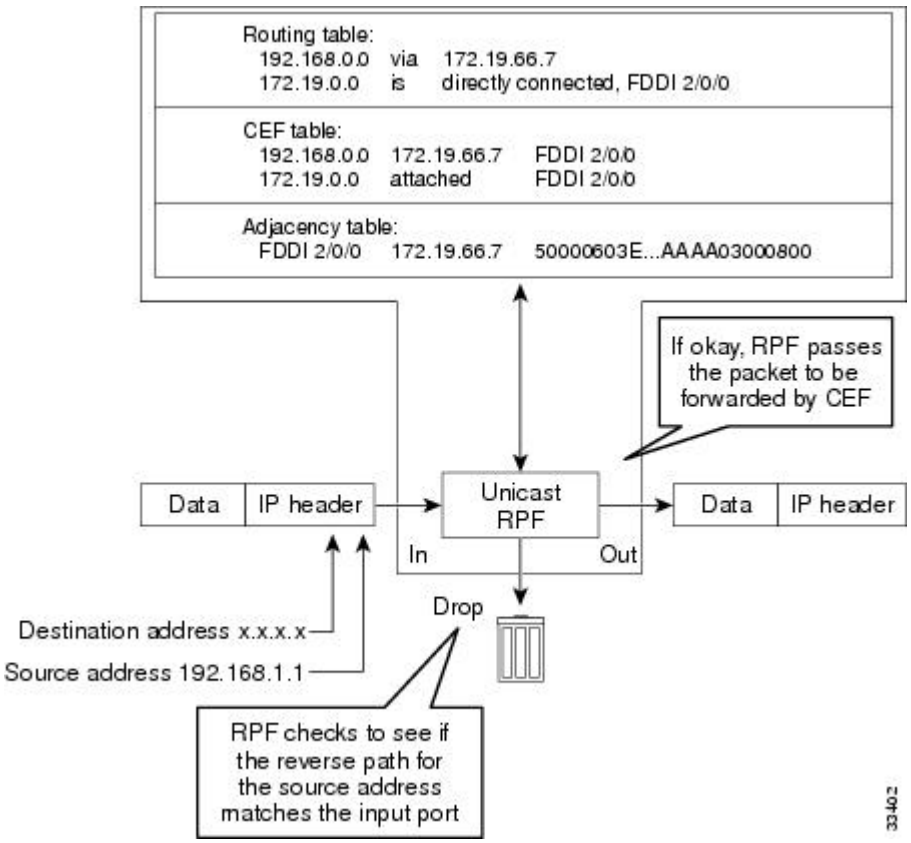
Each time a packet is dropped or forwarded at an interface, that information is counted two ways: globally on the router and at each interface where you have applied Unicast RPF. Global statistics on dropped packets provide information about potential attacks on the network; however, these global statistics do not help to specify which interface is the source of the attack.

Per-interface statistics allow network administrators to track two types of information about malformed packets: Unicast RPF drops and Unicast RPF suppressed drops. Statistics on the number of packets that Unicast RPF drops help to identify the interface that is the entry point of the attack. The Unicast RPF drop count tracks the number of drops at the interface. The Unicast RPF suppressed drop count tracks the number of packets that failed the Unicast RPF check but were forwarded because of the permit permission set up in the ACL. Using the drop count and suppressed drop count statistics, a network administrator can takes steps to isolate the attack at a specific interface.

Note Judicious use of ACL logging can further identify the address or addresses that are being dropped by Unicast RPF.

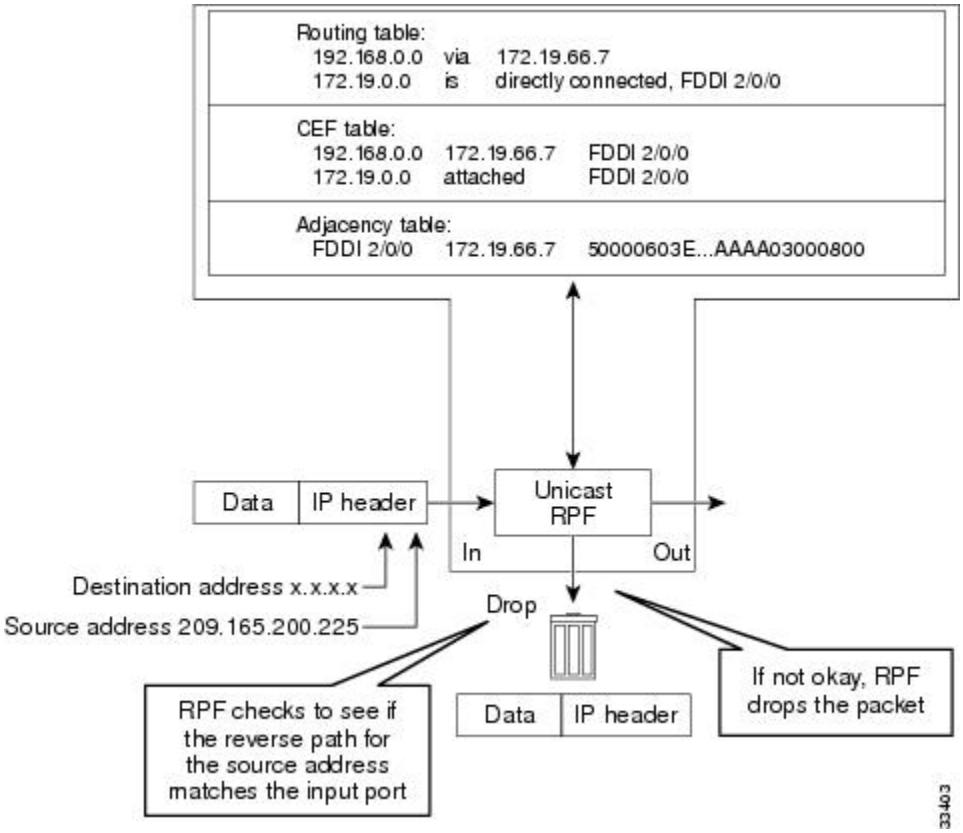
The figure below illustrates how Unicast RPF and CEF work together to validate IP source addresses by verifying packet return paths. In this example, a customer has sent a packet having a source address of 192.168.1.1 from interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 192.168.1.1 has a path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. If there is no matching path, the packet is dropped.

Figure 30: Unicast RPF Validating IP Source Addresses



The figure below illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface FDDI 2/0/0. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to FDDI 2/0/0. If there is a matching path, the packet is forwarded. In this case, there is no reverse entry in the routing table that routes the customer packet back to source address 209.165.200.225 on interface FDDI 2/0/0, and so the packet is dropped.

Figure 31: Unicast RPF Dropping Packets That Fail Verification



How to Configure Unicast Reverse Path Forwarding ACL Support

Configuring Unicast RPF with ACL Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** *ipv6-address/prefix-length*
5. **ipv6 verify unicast source reachable-via** {**rx** | **any**} [*access-list*]
6. **end**
7. **show cef interface** [*type number*]
8. **show ipv6 traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::1/96`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	**ipv6 verify unicast source reachable-via** {**rx** | **any**} [*access-list*] **Example:** `Device(config-if)# ipv6 verify unicast source reachable-via any acl1`	Verifies that a source address exists in the FIB table and enables Unicast RPF.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.
Step 7	**show cef interface** [*type number*] **Example:** `Device# show cef interface gigabitethernet 0/0/1`	Displays detailed Cisco Express Forwarding information for a specified interface or for all interfaces.
Step 8	**show ipv6 traffic** **Example:** `Device# show ipv6 traffic`	Displays statistics about IPv6 traffic.

Example:

The following is sample output from the **show cef interface gigabitethernet 0/0/1** command:

```
Device# show cef interface gigabitethernet 0/0/1

 GigabitEthernet0/0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C67D:4FFF:FEB6:E410
  No Virtual link-local address(es):
  Global unicast address(es):
    2001::1, subnet is 2001::/64
```

```
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FFB6:E410
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  Input features: Verify Unicast Reverse-Path
 IPv6 verify source reachable-via rx, ACL test
   0 verification drop(s) (process), 0 (CEF)
   0 suppressed verification drop(s) (process), 0 (CEF)
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```

The following is sample output from the **show ipv6 traffic** command:

```
Device# show ipv6 traffic

 IPv6 statistics:
        Rcvd:  6 total, 0 local destination
               0 source-routed, 0 truncated
               0 format errors, 0 hop count exceeded
               0 bad header, 0 unknown option, 0 bad source
               0 unknown protocol, 0 not a router
               0 fragments, 0 total reassembled
               0 reassembly timeouts, 0 reassembly failures
        Sent:  34 generated, 28 forwarded
               0 fragmented into 0 fragments, 0 failed
               0 encapsulation failed, 0 no route, 0 too big
               0 RPF drops, 0 RPF suppressed drops
        Mcast: 6 received, 34 sent

ICMP statistics:
        Rcvd: 6 input, 0 checksum errors, 0 too short
              0 unknown info type, 0 unknown error type
              unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
                       0 sa policy, 0 reject route
              parameter: 0 error, 0 header, 0 option
              0 hopcount expired, 0 reassembly timeout,0 too big
              0 echo request, 0 echo reply
              0 group query, 0 group report, 0 group reduce
              0 router solicit, 0 router advert, 0 redirects
              0 neighbor solicit, 0 neighbor advert
        Sent: 34 output, 0 rate-limited
              unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
                       0 sa policy, 0 reject route
              parameter: 0 error, 0 header, 0 option
              0 hopcount expired, 0 reassembly timeout,0 too big
              0 echo request, 0 echo reply
              0 group query, 0 group report, 0 group reduce
              0 router solicit, 18 router advert, 0 redirects
              2 neighbor solicit, 2 neighbor advert
```

Configuration Examples for Unicast Reverse Path Forwarding ACL Support

Example: Configuring Unicast RPF with ACL Support

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# ipv6 address 2001:DB8:1::1/96
Device(config-if)# ipv6 verify unicast source reachable-via any acl1
Device(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Unicast RPF command descriptions	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Cisco Express Forwarding commands	Cisco IOS IP Switching Command Reference

Standards & RFCs

Standard/RFC	Title
RFC 1918	*Address Allocation for Private Internets*
RFC 2267	*Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Unicast Reverse Path Forwarding ACL Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

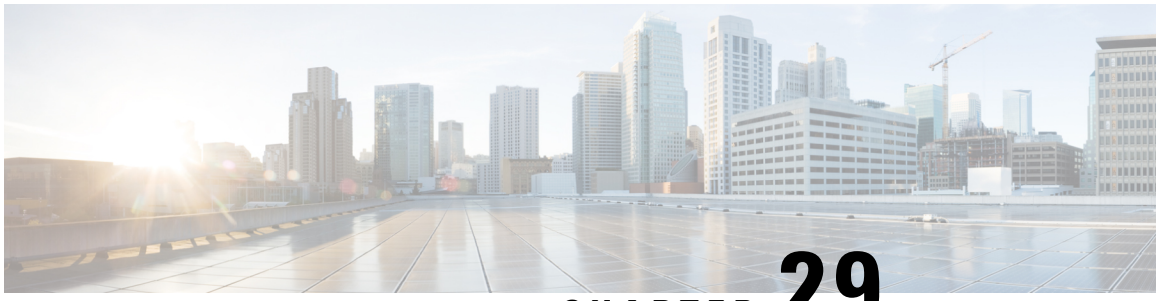
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 36: Feature Information for Unicast Reverse Path Forwarding ACL Support

Feature Name	Releases	Feature Information
Unicast Reverse Path Forwarding ACL Support	Cisco IOS XE Release 3.7S	The Unicast Reverse Path Forwarding feature helps to mitigate problems that are caused by malformed or forged IP source addresses that pass through a device. The Unicast Reverse Path Forwarding ACL support feature adds the ACL support to the Unicast Reverse Path Forwarding feature. With the ACL support, Unicast RPF can determine whether to drop or to forward data packets that have malformed or forged IP source addresses. The following commands were introduced or modified: **ip verify unicast source reachable-via** and **ipv6 verify unicast source reachable-via**.

PART II

BFD



CHAPTER 29

Bidirectional Forwarding Detection

This document describes how to enable the Bidirectional Forwarding Detection (BFD) protocol. BFD is a detection protocol that is designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. It includes a description of how to configure multihop BFD sessions.

BFD provides a consistent failure detection method for network administrators, in addition to fast forwarding path failure detection. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Bidirectional Forwarding Detection

- Cisco Express Forwarding and IP routing must be enabled on all participating routers.
- One of the IP routing protocols supported by BFD must be configured on the routers before BFD is deployed. You should implement fast convergence for the routing protocol that you are using. See the IP routing documentation for your version of Cisco IOS software for information on configuring fast

convergence. See the Restrictions for Bidirectional Forwarding Detection section for more information on BFD routing protocol support in Cisco IOS software.

Restrictions for Bidirectional Forwarding Detection

- When BFD is enabled on an interface, an ACL with "log" option is not supported on that interface.
- The Cisco IOS software incorrectly allows configuration of BFD on virtual-template and dialer interfaces; however, BFD functionality on virtual-template and dialer interfaces is not supported. Avoid configuring BFD on virtual-template and dialer interfaces.
- BFD is supported on point-to-point IPsec tunnel.
- BFD support is not available for all platforms and interfaces. To confirm BFD support for a specific platform or interface and obtain the most accurate platform and hardware restrictions, see the Cisco IOS software release notes for your software version.
- BFD packets are not matched in the QoS policy for self-generated packets.
- BFD packets are matched in the **class class-default** command. So, the user must make sure of the availability of appropriate bandwidth to prevent dropping of BFD packets due to oversubscription.
- BFD between peers goes down when the entry for the BFD control packets in the applied interface ACL has log keyword added as shown in the below example:

  ```
  10 permit ip 10.255.255.0 0.0.0.255 10.255.255.0 0.0.0.255 log
  ```

  This behavior is seen both in echo and nonecho mode, with BFD templates also. Change in timers does not change the behavior. Any value below 750 milliseconds makes the BFD go down, 750 milliseconds 1000 milliseconds results in constant flapping of BFD and from 1000 milliseconds.
- Users have to destroy BFD session on both shut down and no shut down interfaces when a switch happens on the BFD Echo and None-Echo mode.
- The use of echo mode for single-hop BFD sessions on unnumbered interfaces is unreliable and may result in inability to properly detect failures. It is strongly recommended that echo mode be disabled when using single hop BFD on unnumbered interfaces. See Example: Disabling Echo Mode When Configuring Single-Hop BFD on Unnumbered Interfaces, on page 426.
- When configuring BFD over Bundle Interface, the BFD timer should be larger than 750*3 milliseconds and the carrier delay time must be configured as 0 on the physical interface, using the “carrier-delay 0”command in interface configuration.

Support for Point-to-Point IPv4, IPv6, and GRE Tunnels

Depending on your release, Cisco software supports BFD forwarding on point-to-point IPv4, IPv6, and generic routing encapsulation (GRE) tunnels.

Only numbered interfaces are allowed. When the tunnel type is changed from a supported tunnel type to an unsupported one, BFD sessions are brought down for that tunnel and the BFD configuration is removed from the interface.

BFD detection time depends on the topology and infrastructure. For a single-hop IP tunnel that is deployed across physically adjacent devices, the 150 ms (that is, a hello interval of 50 ms with up to three retries)

detection rate applies. However, when the source and destination endpoints of the tunnel are not connected back-to-back, the 150 ms detection rate is not guaranteed.

BFD uses the IP address configured on the tunnel interface. It does not use the tunnel source and destination addresses.

BFD support on DMVPN

- NHRP currently acts only on BFD down events and not on up events.
- Both peers must configure BFD to get BFD support. If one of the peers is not configured with BFD, the other peer creates BFD sessions in down or unknown state.
- BFD intervals configured on the peers should be the same in the BFD echo mode for spoke to spoke refresh to work as expected.

Note NOTE - From Cisco IOS XE 17.11.1a, there is a new keyword delete for the ip nhrp bfd command. This keyword deletes the tunnel entry immediately on a BFD down event and changes the default behavior of deleting the tunnel entry after the expiry of the entry.

Information About Bidirectional Forwarding Detection

BFD Operation

BFD provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent routers, including the interfaces, data links, and forwarding planes.

BFD is a detection protocol that is enabled at the interface and protocol levels. Cisco supports BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between routers. Therefore, in order for a BFD session to be created, BFD must be configured on both systems (or BFD peers). Once BFD has been enabled on the interfaces and at the router level for the appropriate protocols (NHRP and the routing protocol on overlay), a BFD session is created, BFD timers are negotiated, and the BFD peers will begin to send BFD control packets to each other at the negotiated interval.

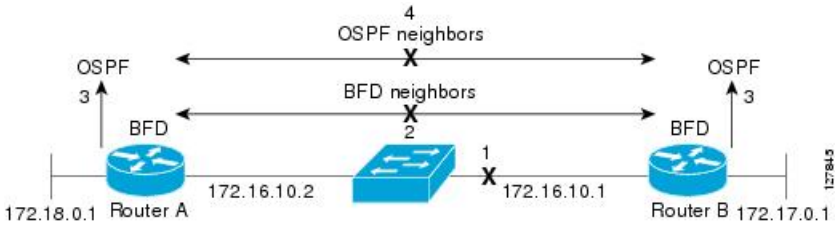
Note To enable BFD, it is recommended to use the BFD template configuration and enable the same BFD template under the interface, instead of directly configuring the BFD parameters under the interface.

Neighbor Relationships

BFD provides fast BFD peer failure detection times independently of all media types, encapsulations, topologies, and routing protocols BGP, EIGRP, IS-IS, and OSPF. By sending rapid failure detection notices to the routing protocols in the local router to initiate the routing table recalculation process, BFD contributes to greatly reduced overall network convergence time. The figure below shows a simple network with two routers running OSPF and BFD. When OSPF discovers a neighbor (1) it sends a request to the local BFD process to initiate

a BFD neighbor session with the OSPF neighbor router (2). The BFD neighbor session with the OSPF neighbor router is established (3).

The figure below shows what happens when a failure occurs in the network (1). The BFD neighbor session with the OSPF neighbor router is torn down (2). BFD notifies the local OSPF process that the BFD neighbor is no longer reachable (3). The local OSPF process tears down the OSPF neighbor relationship (4). If an alternative path is available, the routers will immediately start converging on it.



A routing protocol needs to register with BFD for every neighbor it acquires. Once a neighbor is registered, BFD initiates a session with the neighbor if a session does not already exist.

OSPF registers with BFD when:

- A neighbor finite state machine (FSM) transitions to full state.
- Both OSPF BFD and BFD are enabled.

On broadcast interfaces, OSPF establishes a BFD session only with the designated router (DR) and backup designated router (BDR), but not between any two routers in DROTHER state.

BFD Detection of Failures

Once a BFD session has been established and timer negations are complete, BFD peers send BFD control packets that act in the same manner as an IGP hello protocol to detect liveliness, except at a more accelerated rate. The following information should be noted:

- BFD is a forwarding path failure detection protocol. BFD detects a failure, but the routing protocol must take action to bypass a failed peer.

BFD Version Interoperability

All BFD sessions come up as Version 1 by default and will be interoperable with Version 0. The system automatically performs BFD version detection, and BFD sessions between neighbors will run in the highest common BFD version between neighbors. For example, if one BFD neighbor is running BFD Version 0 and the other BFD neighbor is running Version 1, the session will run BFD Version 0. The output from the **show bfd neighbors** [**details**] command will verify which BFD version a BFD neighbor is running.

See the Example Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default for an example of BFD version detection.

BFD Support for Nonbroadcast Media Interfaces

The **bfd interval** command must be configured on the interface to initiate BFD monitoring.

BFD Support for VPN Routing and Forwarding Interfaces

The BFD feature is extended

to be VPN Routing and Forwarding (VRF) aware to provide fast detection of routing protocol failures between provider edge (PE) and customer edge (CE) routers.

BFD Support for Nonstop Forwarding with Stateful Switchover

Typically, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Nonstop forwarding (NSF) helps to suppress routing flaps in devices that are enabled with stateful switchover (SSO), thereby reducing network instability.

NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored after a switchover. With NSF, peer networking devices do not experience routing flaps. Data traffic is forwarded through intelligent line cards or dual forwarding processors while the standby RP assumes control from the failed active RP during a switchover. The ability of line cards and forwarding processors to remain up through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active RP is key to NSF operation.

In devices that support dual RPs, SSO establishes one of the RPs as the active processor; the other RP is designated as the standby processor, and then synchronizes information between them. A switchover from the active to the standby processor occurs when the active RP fails, when it is removed from the networking device, or when it is manually taken down for maintenance.

BFD Support for Stateful Switchover

The BFD protocol provides short-duration detection of failures in the path between adjacent forwarding engines. In network deployments that use dual RP routers or switches (to provide redundancy), the routers have a graceful restart mechanism that protects the forwarding state during a switchover between the active RP and the standby RP.

The dual RPs have variable switchover times that depend on the ability of the hardware to detect a communication failure. When BFD is running on the RP, some platforms are not able to detect a switchover before the BFD protocol times out; these platforms are referred to as slow switchover platforms.

Stateful BFD on the Standby RP

To ensure a successful switchover to the standby RP, the BFD protocol uses checkpoint messages to send session information from the active RP Cisco IOS instance to the standby RP Cisco IOS instance. The session information includes local and remote discriminators, adjacent router timer information, BFD setup information, and session-specific information such as the type of session and the session version. In addition, the BFD protocol sends session creation and deletion checkpoint messages to create or delete a session on the standby RP.

The BFD sessions on the standby RP do not receive or send packets and do not process expired timers. These sessions wait for a switchover to occur and then send packets for any active sessions so that sessions do not time out on adjacent routers.

When the BFD protocol on the standby RP is notified of a switchover it changes its state to active, registers itself with Cisco Express Forwarding so that it can receive packets, and then sends packets for any elements that have expired.

BFD also uses checkpoint messages to ensure that sessions created by clients on the active RP are maintained during a switchover. When a switchover occurs, BFD starts an SSO reclaim timer. Clients must reclaim their sessions within the duration specified by the reclaim timer or else the session is deleted.

BFD Support for Static Routing

Unlike dynamic routing protocols, such as OSPF and BGP, static routing has no method of peer discovery. Therefore, when BFD is configured, the reachability of the gateway is completely dependent on the state of the BFD session to the specified neighbor. Unless the BFD session is up, the gateway for the static route is considered unreachable, and therefore the affected routes will not be installed in the appropriate Routing Information Base (RIB).

For a BFD session to be successfully established, BFD must be configured on the interface on the peer and there must be a BFD client registered on the peer for the address of the BFD neighbor. When an interface is used by dynamic routing protocols, the latter requirement is usually met by configuring the routing protocol instances on each neighbor for BFD. When an interface is used exclusively for static routing, this requirement must be met by configuring static routes on the peers.

If a BFD configuration is removed from the remote peer while the BFD session is in the up state, the updated state of the BFD session is not signaled to IPv4 static. This will cause the static route to remain in the RIB. The only workaround is to remove the IPv4 static BFD neighbor configuration so that the static route no longer tracks BFD session state. Also, if you change the encapsulation type on a serial interface to one that is unsupported by BFD, BFD will be in a down state on that interface. The workaround is to shut down the interface, change to a supported encapsulation type, and then reconfigure BFD.

A single BFD session can be used by an IPv4 static client to track the reachability of next hops through a specific interface. You can assign a BFD group for a set of BFD-tracked static routes. Each group must have one active static BFD configuration, one or more passive BFD configurations, and the corresponding static routes to be BFD-tracked. Nongroup entries are BFD-tracked static routes for which a BFD group is not assigned. A BFD group must accommodate static BFD configurations that can be part of different VRFs. Effectively, the passive static BFD configurations need not be in the same VRF as that of the active configuration.

For each BFD group, there can be only one active static BFD session. You can configure the active BFD session by adding a static BFD configuration and a corresponding static route that uses the BFD configuration. The BFD session in a group is created only when there is an active static BFD configuration and the static route that uses the static BFD configuration. When the active static BFD configuration or the active static route is removed from a BFD group, all the passive static routes are withdrawn from the RIB. Effectively, all the passive static routes are inactive until an active static BFD configuration and a static route to be tracked by the active BFD session are configured in the group.

Similarly, for each BFD group, there can be one or more passive static BFD configurations and their corresponding static routes to be BFD-tracked. Passive static session routes take effect only when the active BFD session state is reachable. Though the active BFD session state of the group is reachable, the passive static route is added to the RIB only if the corresponding interface state is up. When a passive BFD session is removed from a group, it will not affect the active BFD session if one existed, or the BFD group reachability status.

BFD on Multiple Hops

on arbitrary paths, which might span multiple network hops. The BFD Multihop feature provides subsecond forwarding failure detection for a destination more than one hop, and up to 255 hops, away.

A BFD multihop session is set up between a unique source-destination address pair provided by the client. A session can be set up between two endpoints that have IP connectivity.

You must configure the **bfd-template** and **bfd map** commands to create a multihop template and associate it with one or more maps of destinations and associated BFD timers. You can enable authentication and configure a key chain for BFD multihop sessions.

Multi-hop BFD over IPv6 is supported in software mode only.

Benefits of Using BFD for Failure Detection

When you deploy any feature, it is important to consider all the alternatives and be aware of any trade-offs being made.

The closest alternative to BFD in conventional EIGRP, IS-IS, and OSPF deployments is the use of modified failure detection mechanisms for EIGRP, IS-IS, and OSPF routing protocols.

If you set EIGRP hello and hold timers to their absolute minimums, the failure detection rate for EIGRP falls to within a one- to two-second range.

If you use fast hellos for either IS-IS or OSPF, these Interior Gateway Protocol (IGP) protocols reduce their failure detection mechanisms to a minimum of one second.

There are several advantages to implementing BFD over reduced timer mechanisms for routing protocols:

- Although reducing the EIGRP, IS-IS, and OSPF timers can result in minimum detection timer of one to two seconds, BFD can provide failure detection in less than one second.
- Because BFD is not tied to any particular routing protocol, it can be used as a generic and consistent failure detection mechanism for EIGRP, IS-IS, and OSPF.
- Because some parts of BFD can be distributed to the data plane, it can be less CPU-intensive than the reduced EIGRP, IS-IS, and OSPF timers, which exist wholly at the control plane.

Benefits of BFD Support on DMVPN

- Faster detection of link failure.
- In non-crypto deployments, spoke can detect hub failure only after NHRP registration timeout but hub cannot detect a spoke failure until cache on hub expires (even though routing can re-converge much earlier). BFD allows for a very fast detection for such a failure.
- BFD validates the forwarding path between non authoritative sessions, for example, in scenarios where the hub is configured to respond on behalf of the spoke.
- BFD validates end-to-end data path including the tunnel unlike IKE keepalives/DPD that doesn't pass through the tunnel.
- BFD probes can be off-loaded.

There is no special NHRP configuration needed for BFD support on DMVPN, enabling BFD on an NHRP enabled interface suffices. For DMVPN configuration, refer How to Configure Dynamic Multipoint VPN.

How to Configure Bidirectional Forwarding Detection

Configuring BFD Session Parameters on the Interface

The steps in this procedure show how to configure BFD on the interface by setting the baseline BFD session parameters on an interface. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Perform one of the following steps:
   - **ip address** *ipv4-address mask*
   - **ipv6 address** *ipv6-address*/*mask*
4. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 Perform one of the following steps:

- **ip address** *ipv4-address mask*
- **ipv6 address** *ipv6-address*/*mask*

Example:

Configuring an IPv4 address for the interface:

```
Device(config-if)# ip address 10.201.201.1 255.255.255.0
```

Configuring an IPv6 address for the interface:

```
Device(config-if)# ipv6 address 2001:db8:1:1::1/32
```

Configures an IP address for the interface.

Step 4 **end**

Example:

```
Device(config-if)# end
```

Exits interface configuration mode and returns to privileged EXEC mode.

Configuring BFD Support for Dynamic Routing Protocols

You can enable BFD support for dynamic routing protocols at the router level to enable BFD support globally for all interfaces or you can configure BFD on a per-interface basis at the interface level.

This section describes the following procedures:

Configuring BFD Support for BGP

This section describes the procedure for configuring BFD support for BGP so that BGP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD.

Before you begin

BGP must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

Note Output from the **show bfd neighbors details** command shows the configured intervals. The output does not show intervals that were changed because hardware-offloaded BFD sessions were configured with Tx and Rx intervals that are not multiples of 50 ms.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-tag*
4. **neighbor** *ip-address* **fall-over bfd**
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip bgp neighbor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-tag* **Example:** `Router(config)# router bgp tag1`	Specifies a BGP process and enters router configuration mode.
Step 4	**neighbor** *ip-address* **fall-over bfd** **Example:** `Router(config-router)# neighbor 172.16.10.2 fall-over bfd`	Enables BFD support for fallover.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns the router to privileged EXEC mode.
Step 6	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors detail`	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered. **Note** If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the **show bfd neighbors details** command will show the configured intervals, not the changed ones.
Step 7	**show ip bgp neighbor** **Example:** `Router# show ip bgp neighbor`	(Optional) Displays information about BGP and TCP connections to neighbors.

What to Do Next

See the Monitoring and Troubleshooting BFD section for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections.

Configuring BFD Support for EIGRP

This section describes the procedure for configuring BFD support for EIGRP so that EIGRP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. There are two methods for enabling BFD support for EIGRP:

- You can enable BFD for all of the interfaces for which EIGRP is routing by using the **bfd all-interfaces** command in router configuration mode.
- You can enable BFD for a subset of the interfaces for which EIGRP is routing by using the **bfd interface** *type number* command in router configuration mode.

Before you begin

EIGRP must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

Note Output from the **show bfd neighbors details** command shows the configured intervals. The output does not show intervals that were changed because hardware-offloaded BFD sessions were configured with Tx and Rx intervals that are not multiples of 50 ms.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. Do one of the following:
   - **bfd all-interfaces**
   - **bfd interface** *type number*
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip eigrp interfaces** [*type number*] [*as-number*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 123`	Configures the EIGRP routing process and enters router configuration mode.

Step 4	Do one of the following:
	• **bfd all-interfaces**
	• **bfd interface** *type number*
	Example:
	`Router(config-router)# bfd all-interfaces`
	Example:
	`Router(config-router)# bfd interface FastEthernet 6/0`
		Enables BFD globally on all interfaces associated with the EIGRP routing process.
		or
		Enables BFD on a per-interface basis for one or more interfaces associated with the EIGRP routing process.

Step 5	**end** **Example:** `Router(config-router) end`	Exits router configuration mode and returns the router to privileged EXEC mode.
Step 6	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors details`	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered. **Note** If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the **show bfd neighbors details** command will show the configured intervals, not the changed ones.
Step 7	**show ip eigrp interfaces** [*type number*] [*as-number*] [**detail**] **Example:** `Router# show ip eigrp interfaces detail`	(Optional) Displays the interfaces for which BFD support for EIGRP has been enabled.

What to Do Next

See the Monitoring andTroubleshooting BFD section for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections.

Configuring BFD Support for IS-IS

This section describes the procedures for configuring BFD support for IS-IS so that IS-IS is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. There are two methods for enabling BFD support for IS-IS:

- You can enable BFD for all of the interfaces on which IS-IS is supporting IPv4 routing by using the **bfd all-interfaces** command in router configuration mode. You can then disable BFD for one or more of those interfaces using the **isis bfd disable** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which IS-IS is routing by using the **isis bfd** command in interface configuration mode.

To configure BFD support for IS-IS, perform the steps in one of the following sections:

Prerequisites

IS-IS must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces that you want to run BFD sessions to BFD neighbors over must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

Note Output from the **show bfd neighbors details** command shows the configured intervals. The output does not show intervals that were changed because hardware-offloaded BFD sessions were configured with Tx and Rx intervals that are not multiples of 50 ms.

Configuring BFD Support for IS-IS for All Interfaces

To configure BFD on all IS-IS interfaces that support IPv4 routing, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **bfd all-interfaces**
5. **exit**
6. **interface** *type number*
7. **ip router isis** [*tag*]
8. **isis bfd** [**disable**]
9. **end**
10. **show bfd neighbors** [**details**]
11. **show clns interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis tag1`	Specifies an IS-IS process and enters router configuration mode.
Step 4	**bfd all-interfaces** **Example:** `Router(config-router)# bfd all-interfaces`	Enables BFD globally on all interfaces associated with the IS-IS routing process.
Step 5	**exit** **Example:** `Router(config-router)# exit`	(Optional) Returns the router to global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface fastethernet 6/0`	(Optional) Enters interface configuration mode.
Step 7	**ip router isis** [*tag*] **Example:** `Router(config-if)# ip router isis tag1`	(Optional) Enables support for IPv4 routing on the interface.
Step 8	**isis bfd** [**disable**] **Example:** `Router(config-if)# isis bfd`	(Optional) Enables or disables BFD on a per-interface basis for one or more interfaces associated with the IS-IS routing process. **Note** You should use the **disable** keyword only if you enabled BFD on all of the interfaces that IS-IS is associated with using the **bfd all-interfaces** command in router configuration mode.
Step 9	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns the router to privileged EXEC mode.

	Command or Action	Purpose
Step 10	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors details`	(Optional) Displays information that can be used to verify if the BFD neighbor is active and displays the routing protocols that BFD has registered. **Note** If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the **show bfd neighbors details** command will show the configured intervals, not the changed ones.
Step 11	**show clns interface** **Example:** `Router# show clns interface`	(Optional) Displays information that can be used to verify if BFD for IS-IS has been enabled for a specific IS-IS interface that is associated.

What to Do Next

See the Monitoring and Troubleshooting BFD section for more information on monitoring and troubleshooting BFD. If you want to configure only for a specific subset of interfaces, perform the tasks in the Configuring BFD Support for IS-IS for One or More Interfaces section.

Configuring BFD Support for IS-IS for One or More Interfaces

To configure BFD for only one or more IS-IS interfaces, perform the steps in this section.

Note Output from the **show bfd neighbors details** command shows the configured intervals. The output does not show intervals that were changed because hardware-offloaded BFD sessions were configured with Tx and Rx intervals that are not multiples of 50 ms.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip router isis** [*tag*]
5. **isis bfd [disable]**
6. **end**
7. **show bfd neighbors** [**details**]
8. **show clns interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 6/0`	Enters interface configuration mode.
Step 4	**ip router isis** [*tag*] **Example:** `Router(config-if)# ip router isis tag1`	Enables support for IPv4 routing on the interface.
Step 5	**isis bfd [disable]** **Example:** `Router(config-if)# isis bfd`	Enables or disables BFD on a per-interface basis for one or more interfaces associated with the IS-IS routing process. **Note** You should use the **disable** keyword only if you enabled BFD on all of the interfaces that IS-IS is associated with using the **bfd all-interfaces** command in router configuration mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 7	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors details`	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered. **Note** If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the **show bfd neighbors details** command will show the configured intervals, not the changed ones.
Step 8	**show clns interface** **Example:** `Router# show clns interface`	(Optional) Displays information that can help verify if BFD for IS-IS has been enabled for a specific IS-IS interface that is associated.

What to Do Next

See the Monitoring and Troubleshooting BFD section for more information on monitoring and maintaining BFD. If you want to configure BFD support for another routing protocol, see one of the following sections.

Configuring BFD Support for OSPF

This section describes the procedures for configuring BFD support for OSPF so that OSPF is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. You can either configure BFD support for OSPF globally on all interfaces or configure it selectively on one or more interfaces.

There are two methods for enabling BFD support for OSPF:

- You can enable BFD for all of the interfaces for which OSPF is routing by using the **bfd all-interfaces** command in router configuration mode. You can disable BFD support on individual interfaces using the **ip ospf bfd** [**disable**] command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which OSPF is routing by using the **ip ospf bfd** command in interface configuration mode.

See the following sections for tasks for configuring BFD support for OSPF:

Configuring BFD Support for OSPF for All Interfaces

To configure BFD for all OSPF interfaces, perform the steps in this section.

If you do not want to configure BFD on all OSPF interfaces and would rather configure BFD support specifically for one or more interfaces, see the Configuring BFD Support for OSPF for One or More Interfaces section.

Before you begin

OSPF must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

Note Output from the **show bfd neighbors details** command shows the configured intervals. The output does not show intervals that were changed because hardware-offloaded BFD sessions were configured with Tx and Rx intervals that are not multiples of 50 ms.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **bfd all-interfaces** [**strict-mode**]
5. **exit**
6. **interface** *type number*
7. **ip ospf bfd** [**disable**]
8. **end**
9. **show bfd neighbors** [**details**]

10. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 4`	Specifies an OSPF process and enters router configuration mode.
Step 4	**bfd all-interfaces** [**strict-mode**] **Example:** `Router(config-router)# bfd all-interfaces`	Enables BFD globally on all interfaces associated with the OSPF routing process. **[strict-mode]** - BFD session is established in the strict-mode. In the strict-mode, the OSPF session is not established till the BFD session is established.
Step 5	**exit** **Example:** `Router(config-router)# exit`	(Optional) Returns the router to global configuration mode. Enter this command only if you want to perform Step 7 to disable BFD for one or more interfaces.
Step 6	**interface** *type number* **Example:** `Router(config)# interface fastethernet 6/0`	(Optional) Enters interface configuration mode. Enter this command only if you want to perform Step 7 to disable BFD for one or more interfaces.
Step 7	**ip ospf bfd** [**disable**] **Example:** `Router(config-if)# ip ospf bfd disable`	(Optional) Disables BFD on a per-interface basis for one or more interfaces associated with the OSPF routing process. **Note** You should use the **disable** keyword only if you enabled BFD on all of the interfaces that OSPF is associated with using the **bfd all-interfaces** command in router configuration mode.
Step 8	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns the router to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors detail`	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered. **Note** If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the **show bfd neighbors details** command will show the configured intervals, not the changed ones.
Step 10	**show ip ospf** **Example:** `Router# show ip ospf`	(Optional) Displays information that can help verify if BFD for OSPF has been enabled. If BFD is enabled in strict-mode, the command output displays `BFD is enabled in strict mode.`

What to Do Next

See the Monitoring and Troubleshooting BFD section for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections.

Configuring BFD Support for OSPF for One or More Interfaces

To configure BFD on one or more OSPF interfaces, perform the steps in this section.

Before you begin

OSPF must be running on all participating routers.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf bfd** [**disable**] [**strict-mode**]
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 6/0`	Enters interface configuration mode.
Step 4	**ip ospf bfd** [**disable**] [**strict-mode**] **Example:** `Router(config-if)# ip ospf bfd`	Enables or disables BFD on a per-interface basis for one or more interfaces associated with the OSPF routing process. **Note** You should use the **disable** keyword only if you enabled BFD on all of the interfaces that OSPF is associated with using the **bfd all-interfaces** command in router configuration mode. **[strict-mode]** - BFD session is established in the strict-mode. In the strict-mode, the OSPF session is not established till the BFD session is established.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 6	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors details`	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered. **Note** If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the **show bfd neighbors details** command will show the configured intervals, not the changed ones.
Step 7	**show ip ospf** **Example:** `Router# show ip ospf`	(Optional) Displays information that can help verify if BFD support for OSPF has been enabled. If BFD is enabled in strict-mode, the command output displays `BFD is enabled in strict mode`.

What to Do Next

See the Monitoring and Troubleshooting BFD section for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections.

Configuring BFD Support for HSRP

Perform this task to enable BFD support for Hot Standby Router Protocol (HSRP.) Repeat the steps in this procedure for each interface over which you want to run BFD sessions to HSRP peers.

HSRP supports BFD by default. If HSRP support for BFD has been manually disabled, you can reenable it at the router level to enable BFD support globally for all interfaces or on a per-interface basis at the interface level.

Before you begin

- HSRP must be running on all participating routers.
- Cisco Express Forwarding must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
7. **standby bfd**
8. **exit**
9. **standby bfd all-interfaces**
10. **exit**
11. **show standby neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** [**distributed**] **Example:** `Router(config)# ip cef`	Enables Cisco Express Forwarding or distributed Cisco Express Forwarding.
Step 4	**interface** *type number* **Example:**	Enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface FastEthernet 6/0`	
Step 5	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.0.0.11 255.255.255.0`	Configures an IP address for the interface.
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Router(config-if)# standby 1 ip 10.0.0.11`	Activates HSRP.
Step 7	**standby bfd** **Example:** `Router(config-if)# standby bfd`	(Optional) Enables HSRP support for BFD on the interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**standby bfd all-interfaces** **Example:** `Router(config)# standby bfd all-interfaces`	(Optional) Enables HSRP support for BFD on all interfaces.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.
Step 11	**show standby neighbors** **Example:** `Router# show standby neighbors`	(Optional) Displays information about HSRP support for BFD.

What to Do Next

See the Monitoring and Troubleshooting BFD section for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections.

Configuring BFD Support for Static Routing

Perform this task to configure BFD support for static routing. Repeat the steps in this procedure on each BFD neighbor. For more information, see the "Example: Configuring BFD Support for Static Routing" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Perform one of the following steps:
   - **ip address** *ipv4-address mask*
   - **ipv6 address** *ipv6-address*/*mask*
5. **exit**
6. Perform one of the following steps:
   - **ip route static bfd** *interface-type interface-number ip-address* [**group** *group-name* [**passive**]]
   - **ipv6 route static bfd** *interface-type interface-number ip-address* [**unaasosiated**]
7. Perform one of the following steps:
   - **ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*]
   - **ipv6 route** [**vrf** *vrf-name*] *ipv6 prefix*/*mask* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**name** *next-hop-name*] [**track** *number*] [**tag** *tag*]
8. **exit**
9. Perform one of the following steps:
   - **show ip static route**
   - **show ipv6 static**
10. Perform one of the following steps:
    - **show ip static route bfd**
    - **show ipv6 static bfd**
11. **exit**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface
```

Configures an interface and enters interface configuration mode.

Step 4 Perform one of the following steps:

- **ip address** *ipv4-address mask*
- **ipv6 address** *ipv6-address*/*mask*

Example:

Configuring an IPv4 address for the interface:

```
Device(config-if)# ip address 10.201.201.1 255.255.255.0
```

Configuring an IPv6 address for the interface:

```
Device(config-if)# ipv6 address 2001:db8:1:1::1/32
```

Configures an IP address for the interface.

Step 5 **exit**

Example:

```
Device(config-if)# exit
```

Exits interface configuration mode and returns to global configuration mode.

Step 6 Perform one of the following steps:

- **ip route static bfd** *interface-type interface-number ip-address* [**group** *group-name* [**passive**]]
- **ipv6 route static bfd** *interface-type interface-number ip-address* [**unaasosiated**]

Example:

```
Device(config)# ip route static bfd  10.1.1.1 group group1 passive
Device(config)# ipv6 route static bfd TenGigabitEthernet 0/0/7 19:1:1::2
```

Specifies a static route BFD neighbor.

- The *interface-type*, *interface-number*, and *ip-address* arguments are required because BFD support exists only for directly connected neighbors.

Step 7 Perform one of the following steps:

- **ip route** [**vrf** *vrf-name*] *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [**dhcp**] [*distance*] [**name** *next-hop-name*] [**permanent** | **track** *number*] [**tag** *tag*]
- **ipv6 route** [**vrf** *vrf-name*] *ipv6 prefix*/*mask* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**name** *next-hop-name*] [**track** *number*] [**tag** *tag*]

Example:

```
Device(config)# ip route 10.0.0.0 255.0.0.0
```

```
Device(config)# ipv6 route 19:1:1::/64 TenGigabitEthernet0/0/7 19:1:1::2
```

Specifies a static route BFD neighbor.

Step 8 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode and returns to privileged EXEC mode.

Step 9 Perform one of the following steps:

- **show ip static route**
- **show ipv6 static**

Example:

(Optional) Displays static route database information.

Step 10 Perform one of the following steps:

- **show ip static route bfd**
- **show ipv6 static bfd**

Example:

(Optional) Displays information about the static BFD configuration from the configured BFD groups and nongroup entries.

Step 11 **exit**

Example:

```
Device# exit
```

Exits privileged EXEC mode and returns to user EXEC mode.

Configuring BFD Echo Mode

BFD echo mode is enabled by default, but you can disable it such that it can run independently in each direction.

BFD echo mode works with asynchronous BFD. Echo packets are sent by the forwarding engine and forwarded back along the same path in order to perform detection--the BFD session at the other end does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process; therefore, the number of BFD control packets that are sent out between two BFD neighbors is reduced. In addition, because the forwarding engine is testing the forwarding path on the remote (neighbor) system without involving the remote system, there is an opportunity to improve the interpacket delay variance, thereby achieving quicker failure detection times than when using BFD Version 0 with BFD control packets for the BFD session.

Echo mode is described as without asymmetry when it is running on both sides (both BFD neighbors are running echo mode).

Prerequisites

BFD must be running on all participating routers.

Before using BFD echo mode, you must disable the sending of Internet Control Message Protocol (ICMP) redirect messages by entering the **no ip icmp redirects** command, in order to avoid high CPU utilization.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

Restrictions

- BFD echo mode does not work in conjunction with Unicast Reverse Path Forwarding (uRPF) configuration. If both BFD echo mode and uRPF configurations are enabled, the sessions will flap.
- The use of echo mode for single hop BFD sessions on unnumbered interfaces is unreliable and may result in inability to properly detect failures. It is strongly recommended that echo mode be disabled when using single hop BFD on unnumbered interfaces.

Configuring the BFD Slow Timer

The steps in this procedure show how to change the value of the BFD slow timer. Repeat the steps in this procedure for each BFD router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bfd slow-timer** *milliseconds*
4. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Switch> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Switch# configure terminal
```

Enters global configuration mode.

Step 3 **bfd slow-timer** *milliseconds*

Example:

```
Switch(config)# bfd slow-timer 12000
```

Configures the BFD slow timer.

Step 4 **end**

Example:

```
Switch(config)# end
```

Exits global configuration mode and returns the router to privileged EXEC mode.

Disabling BFD Echo Mode Without Asymmetry

The steps in this procedure show how to disable BFD echo mode without asymmetry—no echo packets will be sent by the router, and the router will not forward BFD echo packets that are received from any neighbor routers.

Repeat the steps in this procedure for each BFD router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configure interface**
4. **no bfd echo**
5. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **configure interface**

Example:

```
Router(config)# configure interface
```

Enters interface configuration mode.

Step 4 **no bfd echo**

Example:

```
Router(config-if)# no bfd echo
```

Disables BFD echo mode.

- Use the **no** form to disable BFD echo mode.

Step 5 **end**

Example:

```
Router(config-if)# end
```

Exits global configuration mode and returns to privileged EXEC mode.

Creating and Configuring BFD Templates

You can configure a single-hop template to specify a set of BFD interval values. BFD interval values specified as part of the BFD template are not specific to a single interface. You can configure a multihop template to associate these values with one or more maps of destinations and associated BFD timers. You can enable authentication and configure a key chain for BFD multihop sessions.

Configuring a Single-Hop Template

Perform this task to create a BFD single-hop template and configure BFD interval timers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bfd-template single-hop** *template-name*
4. **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*
5. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **bfd-template single-hop** *template-name*

Example:

```
Router(config)# bfd-template single-hop bfdtemplate1
```

Creates a single-hop BFD template and enters BFD configuration mode.

Step 4 **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*

Example:

```
Router(bfd-config)# interval min-tx 120 min-rx 100 multiplier 3
```

Configures the transmit and receive intervals between BFD packets, and specifies the number of consecutive BFD control packets that must be missed before BFD declares that a peer is unavailable.

Step 5 **end**

Example:

```
Router(bfd-config)# end
```

Exits BFD configuration mode and returns the router to privileged EXEC mode.

Configuring a Multihop Template

Perform this task to create a BFD multihop template and configure BFD interval timers, authentication, and key chain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bfd-template multi-hop** *template-name*
4. **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*
5. **authentication** *authentication-type* **keychain** *keychain-name*
6. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **bfd-template multi-hop** *template-name*

Example:

```
Router(config)# bfd-template multi-hop mh-template1
```

Creates a BFD multihop BFD template and enters BFD configuration mode.

Step 4 **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*

Example:

```
Router(bfd-config)# interval min-tx 120 min-rx 100 multiplier 3
```

Configures the transmit and receive intervals between BFD packets, and specifies the number of consecutive BFD control packets that must be missed before BFD declares that a peer is unavailable.

Step 5 **authentication** *authentication-type* **keychain** *keychain-name*

Example:

```
Router(bfd-config)# authentication keyed-sha-1 keychain bfd-multihop
```

Configures authentication for the multihop template and specifies the authentication type.

Step 6 **end**

Example:

```
Router(bfd-config)# end
```

Exits BFD configuration mode and returns the router to privileged EXEC mode.

Configuring BFD Support on DMVPN

BFD intervals can be directly configured on tunnel interface as shown below:

```
enable
configure terminal
```

```
interface tunnel1
bfd interval 1000 min_rx 1000 multiplier 5
no echo
```

BFD intervals can also be configured by defining a template and attaching it to the tunnel interface as shown below

```
enable
configure terminal
bfd-template single-hop sample
interval min-tx 1000 min-rx 1000 multiplier 5
interface tunnel1
bfd template sample
```

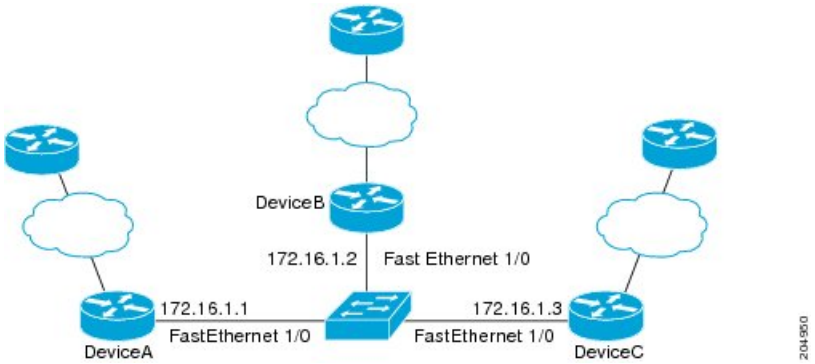
Configuration Examples for Bidirectional Forwarding Detection

Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default

In the following example, the EIGRP network contains RouterA, RouterB, and RouterC. Fast Ethernet interface 1/0 on RouterA is connected to the same network as Fast Ethernet interface 1/0 on Router B. Fast Ethernet interface 1/0 on RouterB is connected to the same network as Fast Ethernet interface 1/0 on RouterC.

RouterA and RouterB are running BFD Version 1, which supports echo mode, and RouterC is running BFD Version 0, which does not support echo mode. The BFD sessions between RouterC and its BFD neighbors are said to be running echo mode with asymmetry because echo mode will run on the forwarding path for RouteA and RouterB, and their echo packets will return along the same path for BFD sessions and failure detections, while their BFD neighbor RouterC runs BFD Version 0 and uses BFD controls packets for BFD sessions and failure detections.

The figure below shows a large EIGRP network with several routers, three of which are BFD neighbors that are running EIGRP as their routing protocol.



The example, starting in global configuration mode, shows the configuration of BFD.

Configuration for RouterA

```
interface Fast Ethernet0/0
 no shutdown
 ip address 10.4.9.14 255.255.255.0
 duplex auto
 speed auto
!
interface Fast Ethernet1/0
 ip address 172.16.1.1 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 3
 no shutdown
 duplex auto
 speed auto
!
router eigrp 11
 network 172.16.0.0
 bfd all-interfaces
 auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
 exec-timeout 30 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
end
```

Configuration for RouterB

```
!
interface Fast Ethernet0/0
 no shutdown
 ip address 10.4.9.34 255.255.255.0
 duplex auto
 speed auto
!
interface Fast Ethernet1/0
 ip address 172.16.1.2 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 3
 no shtdown
 duplex auto
 speed auto
!
router eigrp 11
 network 172.16.0.0
 bfd all-interfaces
 auto-summary
```

```
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
 exec-timeout 30 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
!
end
```

Configuration for RouterC

```
!
!
interface Fast Ethernet0/0
 no shutdown
 ip address 10.4.9.34 255.255.255.0
 duplex auto
 speed auto
!
interface Fast Ethernet1/0
 ip address 172.16.1.3 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 3
 no shutdown
 duplex auto
 speed auto
!
router eigrp 11
 network 172.16.0.0
 bfd all-interfaces
 auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
no ip http server
!
logging alarm informational
!
control-plane
!
line con 0
 exec-timeout 30 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
```

```
!
!
end
```

The output from the **show bfd neighbors details** command from RouterA verifies that BFD sessions have been created among all three routers and that EIGRP is registered for BFD support. The first group of output shows that RouterC with the IP address 172.16.1.3 runs BFD Version 0 and therefore does not use the echo mode. The second group of output shows that RouterB with the IP address 172.16.1.2 does run BFD Version 1, and the 50 millisecond BFD interval parameter had been adopted. The relevant command output is shown in bold in the output.

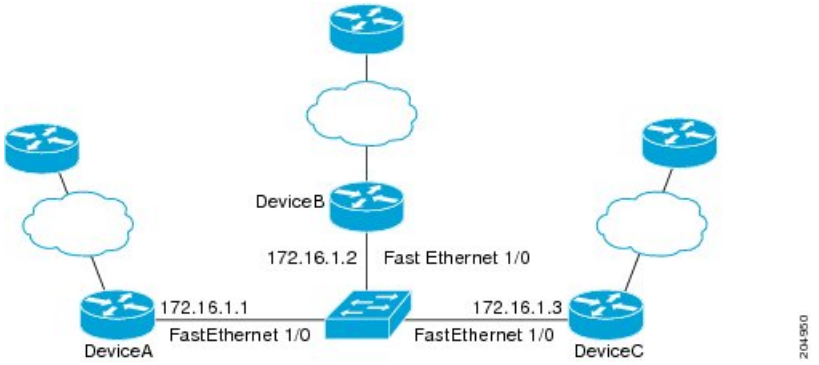
```
RouterA# show bfd neighbors details

OurAddr
       NeighAddr
      LD/RD  RH/RS    Holdown(mult)  State     Int
172.16.1.1    172.16.1.3
     5/3    1(RH)    150 (3 )       Up   Fa1/0
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(1364284)
Rx Count: 1351813, Rx Interval (ms) min/max/avg: 28/64/49 last: 4 ms ago
Tx Count: 1364289, Tx Interval (ms) min/max/avg: 40/68/49 last: 32 ms ago
Registered protocols: EIGRP
Uptime: 18:42:45
Last packet: Version: 0
             - Diagnostic: 0
              I Hear You bit: 1     - Demand bit: 0
              Poll bit: 0           - Final bit: 0
              Multiplier: 3         - Length: 24
              My Discr.: 3          - Your Discr.: 5
              Min tx interval: 50000    - Min rx interval: 50000
              Min Echo interval: 0
OurAddr       NeighAddr
     LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.1    172.16.1.2

    6/1    Up        0    (3 )   Up        Fa1/0
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(317)
Rx Count: 305, Rx Interval (ms) min/max/avg: 1/1016/887 last: 448 ms ago
Tx Count: 319, Tx Interval (ms) min/max/avg: 1/1008/880 last: 532 ms ago
Registered protocols: EIGRP
Uptime: 00:04:30
Last packet: Version: 1

         - Diagnostic: 0
             State bit: Up         - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 1          - Your Discr.: 6
             Min tx interval: 1000000    - Min rx interval: 1000000
             Min Echo interval: 50000
```

The output from the **show bfd neighbors details** command on Router B verifies that BFD sessions have been created and that EIGRP is registered for BFD support. As previously noted, RouterA runs BFD Version 1,

therefore echo mode is running, and RouterC runs BFD Version 0, so echo mode does not run. The relevant command output is shown in bold in the output.

```
RouterB# show bfd neighbors details

OurAddr       NeighAddr
     LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.2    172.16.1.1
    1/6    Up        0    (3 )   Up        Fa1/0
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(337)
Rx Count: 341, Rx Interval (ms) min/max/avg: 1/1008/882 last: 364 ms ago
Tx Count: 339, Tx Interval (ms) min/max/avg: 1/1016/886 last: 632 ms ago
Registered protocols: EIGRP
Uptime: 00:05:00
Last packet: Version: 1
           - Diagnostic: 0
             State bit: Up         - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 6          - Your Discr.: 1
             Min tx interval: 1000000    - Min rx interval: 1000000
             Min Echo interval: 50000
OurAddr       NeighAddr

 LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.2    172.16.1.3
     3/6    1(RH)     118  (3 )   Up        Fa1/0
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(5735)
Rx Count: 5731, Rx Interval (ms) min/max/avg: 32/72/49 last: 32 ms ago
Tx Count: 5740, Tx Interval (ms) min/max/avg: 40/64/50 last: 44 ms ago
Registered protocols: EIGRP
Uptime: 00:04:45
Last packet: Version: 0
            - Diagnostic: 0
              I Hear You bit: 1     - Demand bit: 0
              Poll bit: 0           - Final bit: 0
              Multiplier: 3         - Length: 24
              My Discr.: 6          - Your Discr.: 3
              Min tx interval: 50000    - Min rx interval: 50000
              Min Echo interval: 0
```

The figure below shows that Fast Ethernet interface 1/0 on RouterB has failed. When Fast Ethernet interface 1/0 on RouterB is shut down, the BFD statistics of the corresponding BFD sessions on RouterA and RouterB are reduced.



When Fast Ethernet interface 1/0 on RouterB fails, BFD will no longer detect Router B as a BFD neighbor for RouterA or for RouterC. In this example, Fast Ethernet interface 1/0 has been administratively shut down on RouterB.

The following output from the **show bfd neighbors** command on RouterA now shows only one BFD neighbor for RouterA in the EIGRP network. The relevant command output is shown in bold in the output.

```
RouterA# show bfd neighbors
OurAddr       NeighAddr

    LD/RD  RH/RS   Holdown(mult)  State     Int
172.16.1.1    172.16.1.3

    5/3    1(RH)     134  (3 )   Up        Fa1/0
```

The following output from the **show bfd neighbors** command on RouterC also now shows only one BFD neighbor for RouterC in the EIGRP network. The relevant command output is shown in bold in the output.

```
RouterC# show bfd neighbors

OurAddr       NeighAddr

   LD/RD RH  Holdown(mult)  State     Int
172.16.1.3    172.16.1.1

    3/5  1   114  (3 )      Up        Fa1/0
```

Example: Configuring BFD in an OSPF Network

In the following example, the simple OSPF network consists of Router A and Router B. Fast Ethernet interface 0/1 on Router A is connected to the same network as Fast Ethernet interface 6/0 in Router B. The example, starting in global configuration mode, shows the configuration of BFD. For both Routers A and B, BFD is configured globally for all interfaces associated with the OSPF process.

Configuration for Router A

```
!
interface Fast Ethernet 0/1
 ip address 172.16.10.1 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 3
!
interface Fast Ethernet 3/0.1
ip address 172.17.0.1 255.255.255.0
```

```
!
router ospf 123
 log-adjacency-changes detail
 network 172.16.0.0 0.0.0.255 area 0
 network 172.17.0.0 0.0.0.255 area 0
 bfd all-interfaces
```

Configuration for Router B

```
!
interface Fast Ethernet 6/0
 ip address 172.16.10.2 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 3
!
interface Fast Ethernet 6/1
 ip address 172.18.0.1 255.255.255.0
!
router ospf 123
 log-adjacency-changes detail
 network 172.16.0.0 0.0.255.255 area 0
 network 172.18.0.0 0.0.255.255 area 0
 bfd all-interfaces
```

The output from the **show bfd neighbors details** command verifies that a BFD session has been created and that OSPF is registered for BFD support. The relevant command output is shown in bold in the output.

Router A

```
RouterA# show bfd neighbors details
OurAddr       NeighAddr     LD/RD RH  Holdown(mult)  State     Int
172.16.10.1   172.16.10.2    1/2  1   532  (3 )      Up        Fa0/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 1000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332 last: 68 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196 last: 192 ms ago
Registered protocols: OSPF


Uptime: 02:18:49
Last packet: Version: 0
            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 2          - Your Discr.: 1
             Min tx interval: 50000    - Min rx interval: 1000
             Min Echo interval: 0
```

The output from the **show bfd neighbors details** command from the line card on Router B verifies that a BFD session has been created:

Router B

```
RouterB# attach 6
Entering Console for 8 Port Fast Ethernet in Slot: 6
Type "exit" to end this session
Press RETURN to get started!
Router> show bfd neighbors details
```

```
Cleanup timer hits: 0
OurAddr       NeighAddr     LD/RD RH  Holdown(mult)  State     Int
172.16.10.2   172.16.10.1    8/1  1   1000 (5 )      Up        Fa6/0
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 1000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 1000(0), Hello (hits): 200(5995)
Rx Count: 10126, Rx Interval (ms) min/max/avg: 152/248/196 last: 0 ms ago
Tx Count: 5998, Tx Interval (ms) min/max/avg: 204/440/332 last: 12 ms ago
Last packet: Version: 0            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 5         - Length: 24
             My Discr.: 1          - Your Discr.: 8
             Min tx interval: 200000    - Min rx interval: 200000
             Min Echo interval: 0
Uptime: 00:33:13
SSO Cleanup Timer called: 0
SSO Cleanup Action Taken: 0
Pseudo pre-emptive process count: 239103 min/max/avg: 8/16/8 last: 0 ms ago
 IPC Tx Failure Count: 0
 IPC Rx Failure Count: 0
 Total Adjs Found: 1
```

The output of the **show ip ospf** command verifies that BFD has been enabled for OSPF. The relevant command output is shown in bold in the output.

Router A

```
RouterA# show ip ospf

 Routing Process "ospf 123" with ID 172.16.10.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 External flood list length 0
 BFD is enabled

    Area BACKBONE(0)
        Number of interfaces in this area is 2 (1 loopback)
        Area has no authentication
        SPF algorithm last executed 00:00:08.828 ago
        SPF algorithm executed 9 times
        Area ranges are
        Number of LSA 3. Checksum Sum 0x028417
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
```

```
        Number of DoNotAge LSA 0
        Flood list length 0
```

Router B

```
RouterB# show ip ospf

 Routing Process "ospf 123" with ID 172.18.0.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x0
 Number of opaque AS LSA 0. Checksum Sum 0x0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
 BFD is enabled

    Area BACKBONE(0)
        Number of interfaces in this area is 2 (1 loopback)
        Area has no authentication
        SPF algorithm last executed 02:07:30.932 ago
        SPF algorithm executed 7 times
        Area ranges are
        Number of LSA 3. Checksum Sum 0x28417
        Number of opaque link LSA 0. Checksum Sum 0x0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

The output of the **show ip ospf interface** command verifies that BFD has been enabled for OSPF on the interfaces connecting Router A and Router B. The relevant command output is shown in bold in the output.

Router A

```
RouterA# show ip ospf interface Fast Ethernet 0/1
show ip ospf interface Fast Ethernet 0/1
Fast Ethernet0/1 is up, line protocol is up
  Internet Address 172.16.10.1/24, Area 0
  Process ID 123, Router ID 172.16.10.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1, BFD enabled
  Designated Router (ID) 172.18.0.1, Interface address 172.16.10.2
  Backup Designated router (ID) 172.16.10.1, Interface address 172.16.10.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:03
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
```

```
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 172.18.0.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

Router B

```
RouterB# show ip ospf interface Fast Ethernet 6/1
Fast Ethernet6/1 is up, line protocol is up
  Internet Address 172.18.0.1/24, Area 0
  Process ID 123, Router ID 172.18.0.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1, BFD enabled
  Designated Router (ID) 172.18.0.1, Interface address 172.18.0.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

Example: Configuring BFD in a BGP Network

In the following example, the simple BGP network consists of Router A and Router B. Fast Ethernet interface 0/1 on Router A is connected to the same network as Fast Ethernet interface 6/0 in Router B. The example, starting in global configuration mode, shows the configuration of BFD.

Configuration for Router A

```
!
interface Fast Ethernet 0/1
 ip address 172.16.10.1 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 3
!
interface Fast Ethernet 3/0.1
ip address 172.17.0.1 255.255.255.0
!
!
router bgp 40000
 bgp log-neighbor-changes
 neighbor 172.16.10.2 remote-as 45000
 neighbor 172.16.10.2 fall-over bfd
 !
 address-family ipv4
 neighbor 172.16.10.2 activate
 no auto-summary
 no synchronization
 network 172.18.0.0 mask 255.255.255.0
 exit-address-family
!
```

Configuration for Router B

```
!
interface Fast Ethernet 6/0
 ip address 172.16.10.2 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 3
!
interface Fast Ethernet 6/1
 ip address 172.18.0.1 255.255.255.0
!
router bgp 45000
 bgp log-neighbor-changes
 neighbor 172.16.10.1 remote-as 40000
 neighbor 172.16.10.1 fall-over bfd
 !
 address-family ipv4
 neighbor 172.16.10.1 activate
 no auto-summary
 no synchronization
 network 172.17.0.0 mask 255.255.255.0
 exit-address-family
!
```

The output from the **show bfd neighbors details** command from Router A verifies that a BFD session has been created and that BGP is registered for BFD support. The relevant command output is shown in bold in the output.

Router A

```
RouterA# show bfd neighbors details

OurAddr       NeighAddr     LD/RD RH  Holdown(mult)  State     Int
172.16.10.1   172.16.10.2    1/8  1    332  (3 )      Up        Fa0/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 1000, Received Multiplier: 3
Holdown (hits): 600(0), Hello (hits): 200(15491)
Rx Count: 9160, Rx Interval (ms) min/max/avg: 200/440/332 last: 268 ms ago
Tx Count: 15494, Tx Interval (ms) min/max/avg: 152/248/197 last: 32 ms ago
Registered protocols: BGP
Uptime: 00:50:45
Last packet: Version: 0            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 8          - Your Discr.: 1
             Min tx interval: 50000    - Min rx interval: 1000
             Min Echo interval: 0
```

The output from the **show bfd neighbors details** command from the line card on Router B verifies that a BFD session has been created:

Router B

```
RouterB# attach 6
Entering Console for 8 Port Fast Ethernet in Slot: 6
Type "exit" to end this session
Press RETURN to get started!
Router> show bfd neighbors details
Cleanup timer hits: 0
```

```
OurAddr       NeighAddr     LD/RD RH  Holdown(mult)  State     Int
172.16.10.2   172.16.10.1    8/1  1   1000 (5 )      Up        Fa6/0
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 1000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 1000(0), Hello (hits): 200(5995)
Rx Count: 10126, Rx Interval (ms) min/max/avg: 152/248/196 last: 0 ms ago
Tx Count: 5998, Tx Interval (ms) min/max/avg: 204/440/332 last: 12 ms ago
Last packet: Version: 0            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 5         - Length: 24
             My Discr.: 1          - Your Discr.: 8
             Min tx interval: 200000    - Min rx interval: 200000
             Min Echo interval: 0
Uptime: 00:33:13
SSO Cleanup Timer called: 0
SSO Cleanup Action Taken: 0
Pseudo pre-emptive process count: 239103 min/max/avg: 8/16/8 last: 0 ms ago
 IPC Tx Failure Count: 0
 IPC Rx Failure Count: 0
 Total Adjs Found: 1
```

The output of the **show ip bgp neighbors** command verifies that BFD has been enabled for the BGP neighbors:

Router A

```
RouterA# show ip bgp neighbors
BGP neighbor is 172.16.10.2,  remote AS 45000, external link
 Using BFD to detect fast fallover
.
.
.
```

Router B

```
RouterB# show ip bgp neighbors
BGP neighbor is 172.16.10.1,  remote AS 40000, external link
 Using BFD to detect fast fallover
.
.
.
```

Example: Configuring BFD in an IS-IS Network

In the following example, the simple IS-IS network consists of Router A and Router B. Fast Ethernet interface 0/1 on Router A is connected to the same network as Fast Ethernet interface 6/0 for Router B. The example, starting in global configuration mode, shows the configuration of BFD.

Configuration for Router A

```
!
interface Fast Ethernet 0/1
 ip address 172.16.10.1 255.255.255.0
ip router isis
 bfd interval 50 min_rx 50 multiplier 3
!
interface Fast Ethernet 3/0.1
```

```
ip address 172.17.0.1 255.255.255.0
ip router isis
!
router isis
 net 49.0001.1720.1600.1001.00
 bfd all-interfaces
!
```

Configuration for Router B

```
!
interface Fast Ethernet 6/0
 ip address 172.16.10.2 255.255.255.0
ip router isis
 bfd interval 50 min_rx 50 multiplier 3
!
interface Fast Ethernet 6/1
 ip address 172.18.0.1 255.255.255.0
ip router isis
!
router isis
 net 49.0000.0000.0002.00
 bfd all-interfaces
!
```

The output from the **show bfd neighbors details** command from Router A verifies that a BFD session has been created and that IS-IS is registered for BFD support:

```
RouterA# show bfd neighbors details

OurAddr       NeighAddr     LD/RD RH  Holdown(mult)  State     Int
172.16.10.1   172.16.10.2    1/8  1   536  (3 )      Up        Fa0/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 1000, Received Multiplier: 3
Holdown (hits): 600(0), Hello (hits): 200(23543)
Rx Count: 13877, Rx Interval (ms) min/max/avg: 200/448/335 last: 64 ms ago
Tx Count: 23546, Tx Interval (ms) min/max/avg: 152/248/196 last: 32 ms ago
Registered protocols: ISIS
Uptime: 01:17:09
Last packet: Version: 0            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 3         - Length: 24
             My Discr.: 8          - Your Discr.: 1
             Min tx interval: 50000    - Min rx interval: 1000
             Min Echo interval: 0
```

The output from the **show bfd neighbors details** command from the line card on Router B verifies that a BFD session has been created:

```
RouterB# attach 6

Entering Console for 8 Port Fast Ethernet in Slot: 6
Type "exit" to end this session
Press RETURN to get started!
Router> show bfd neighbors details
Cleanup timer hits: 0
OurAddr       NeighAddr     LD/RD RH  Holdown(mult)  State     Int
172.16.10.2   172.16.10.1    8/1  1   1000 (5 )      Up        Fa6/0
Local Diag: 0, Demand mode: 0, Poll bit: 0
```

```
MinTxInt: 50000, MinRxInt: 1000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 1000(0), Hello (hits): 200(5995)
Rx Count: 10126, Rx Interval (ms) min/max/avg: 152/248/196 last: 0 ms ago
Tx Count: 5998, Tx Interval (ms) min/max/avg: 204/440/332 last: 12 ms ago
Last packet: Version: 0            - Diagnostic: 0
             I Hear You bit: 1     - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 5         - Length: 24
             My Discr.: 1          - Your Discr.: 8
             Min tx interval: 200000    - Min rx interval: 200000
             Min Echo interval: 0
Uptime: 00:33:13
SSO Cleanup Timer called: 0
SSO Cleanup Action Taken: 0
Pseudo pre-emptive process count: 239103 min/max/avg: 8/16/8 last: 0 ms ago
 IPC Tx Failure Count: 0
 IPC Rx Failure Count: 0
 Total Adjs Found: 1
```

Example: Configuring BFD in an HSRP Network

In the following example, the HSRP network consists of Router A and Router B. Fast Ethernet interface 2/0 on Router A is connected to the same network as Fast Ethernet interface 2/0 on Router B. The example, starting in global configuration mode, shows the configuration of BFD.

Note In the following example, the **standby bfd** and the **standby bfd all-interfaces** commands are not displayed. HSRP support for BFD peering is enabled by default when BFD is configured on the router or interface using the **bfd interval** command. The **standby bfd**and **standby bfd all-interfaces** commands are needed only if BFD has been manually disabled on a router or interface.

Router A

```
ip cef
interface Fast Ethernet2/0
 no shutdown
 ip address 10.0.0.2 255.0.0.0
 ip router-cache cef
 bfd interval 200 min_rx 200 multiplier 3
 standby 1 ip 10.0.0.11
 standby 1 preempt
 standby 1 priority 110

 standby 2 ip 10.0.0.12
 standby 2 preempt
 standby 2 priority 110
```

Router B

```
interface Fast Ethernet2/0
 ip address 10.1.0.22 255.255.0.0
 no shutdown
 bfd interval 200 min_rx 200 multiplier 3
 standby 1 ip 10.0.0.11
 standby 1 preempt
```

```
 standby 1 priority 90
 standby 2 ip 10.0.0.12
 standby 2 preempt
 standby 2 priority 80
```

The output from the **show standby neighbors** command verifies that a BFD session has been created:

```
RouterA#show standby neighbors

HSRP neighbors on Fast Ethernet2/0
  10.1.0.22
    No active groups
    Standby groups: 1
    BFD enabled !
RouterB# show standby neighbors

HSRP neighbors on Fast Ethernet2/0
  10.0.0.2
    Active groups: 1
    No standby groups
    BFD enabled !
```

Example: Configuring BFD Support for Static Routing

In the following example, the network consists of Device A and Device B. Serial interface 2/0 on Device A is connected to the same network as serial interface 2/0 on Device B. In order for the BFD session to come up, Device B must be configured.

Device A

```
configure terminal
interface Serial 2/0
ip address 10.201.201.1 255.255.255.0
bfd interval 500 min_rx 500 multiplier 5
ip route static bfd Serial 2/0 10.201.201.2
ip route 10.0.0.0 255.0.0.0 Serial 2/0 10.201.201.2
```

Device B

```
configure terminal
interface Serial 2/0
ip address 10.201.201.2 255.255.255.0
bfd interval 500 min_rx 500 multiplier 5
ip route static bfd Serial 2/0 10.201.201.1
ip route 10.1.1.1 255.255.255.255 Serial 2/0 10.201.201.1
```

Note that the static route on Device B exists solely to enable the BFD session between 10.201.201.1 and 10.201.201.2. If there is no useful static route that needs to be configured, select a prefix that will not affect packet forwarding, for example, the address of a locally configured loopback interface.

In the following example, there is an active static BFD configuration to reach 209.165.200.225 through Ethernet interface 0/0 in the BFD group testgroup. As soon as the static route is configured that is tracked by the configured static BFD, a single hop BFD session is initiated to 209.165.200.225 through Ethernet interface 0/0. The prefix 10.0.0.0/8 is added to the RIB if a BFD session is successfully established.

```
configure terminal
```

```
ip route static bfd Ethernet 0/0 209.165.200.225 group testgroup
ip route 10.0.0.0 255.255.255.224 Ethernet 0/0 209.165.200.225
```

In the following example, a BFD session to 209.165.200.226 through Ethernet interface 0/0.1001 is marked to use the group testgroup. That is, this configuration is a passive static BFD. Though there are static routes to be tracked by the second static BFD configuration, a BFD session is not triggered for 209.165.200.226 through Ethernet interface 0/0.1001. The existence of the prefixes 10.1.1.1/8 and 10.2.2.2/8 is controlled by the active static BFD session (Ethernet interface 0/0 209.165.200.225).

```
configure terminal
ip route static bfd Ethernet 0/0 209.165.200.225 group testgroup
ip route 10.0.0.0 255.255.255.224 Ethernet 0/0 209.165.200.225
ip route static bfd Ethernet 0/0.1001 209.165.200.226 group testgroup passive
ip route 10.1.1.1 255.255.255.224 Ethernet 0/0.1001 209.165.200.226
ip route 10.2.2.2 255.255.255.224 Ethernet 0/0.1001 209.165.200.226
```

Example: BFD Support on DMVPN

Example: BFD Support on DMVPN

The following is an example of configuring BFD support on DMVPN on hub.

```
bfd-template single-hop sample
 interval min-tx 1000 min-rx 1000 multiplier 5
!
interface Tunnel0
 ip address 10.0.0.1 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco123
 ip nhrp network-id 5
 ip nhrp redirect
 ip mtu 1400
 ip tcp adjust-mss 1360
 bfd template sample
 tunnel source GigabitEthernet0/0/0
 tunnel mode gre multipoint
 tunnel key 6
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.0.0.0
 negotiation auto
!
router eigrp 2
network 10.0.0.0 0.0.0.255
bfd all-interfaces
auto-summary
!
```

The following is an example of configuring BFD support on DMVPN on spoke.

```
bfd-template single-hop sample
 interval min-tx 1000 min-rx 1000 multiplier 5
!
interface Tunnel1
 ip address 10.0.0.10 255.255.255.0
```

```
 no ip redirects
 ip nhrp authentication cisco123
 ip nhrp network-id 5
 ip nhrp nhs 10.0.0.1 nbma 10.0.0.10 multicast
 bfd template sample
 tunnel source GigabitEthernet0/0/0
 tunnel mode gre multipoint
 tunnel key 6
!
interface GigabitEthernet0/0/0
 mtu 4000
 ip address 11.0.0.1 255.0.0.0
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/0/1
 mtu 6000
 ip address 111.0.0.1 255.255.255.0
 negotiation auto
!
router eigrp 2
 network 11.0.0.0 0.0.0.255
 network 111.0.0.0 0.0.0.255
 network 10.0.0.0 0.0.0.255
bfd all-interfaces
auto-summary
!
ip route 0.0.0.0 0.0.0.0 10.0.0.2
```

The following example outlines how to delete the tunnel entry details when BFD support is down by addition of ip nhrp bfd command. By default, the tunnel entry is not immediately deleted and is deleted after expiry of the entry.

```
!
interface Tunnel0
ip address 10.0.1.100 255.255.255.0
no ip redirects
ip nhrp authentication testing
ip nhrp summary-map 192.168.0.0/16 72.68.100.2
ip nhrp summary-map 77.77.0.0/16 72.68.100.2
ip nhrp network-id 100
ip nhrp bfd delete
ip nhrp redirect
bfd interval 1000 min_rx 1000 multiplier 5
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel key 100
tunnel protection ipsec profile default
!
```

Note In this configuration, the tunnel entry is immediately deleted upon receiving a BFD down event. Without this configuration, the cache entry pertaining to the tunnel address of the peer is not deleted and performs its default behaviour.

The following is an example to illustrate faster convergence on spoke.

```
interface Tunnel1
ip address 18.0.0.10 255.255.255.0
```

```
no ip redirects
ip nhrp authentication cisco123
ip nhrp network-id 12
ip nhrp nhs 10.0.0.1 nbma 10.0.0.10 multicast
bfd template sample
tunnel source GigabitEthernet0/0/0
tunnel mode gre multipoint
tunnel key 18
tunnel protection ipsec profile MY_PROFILE
!
bfd-template single-hop sample
interval min-tx 1000 min-rx 1000 multiplier 3
echo
!
router eigrp 2
bfd interface Tunnel1    ------------------------> Specify the interface on which the routing
 protocol must act for BFD up/down events
network 11.0.0.0 0.0.0.255
network 111.0.0.0 0.0.0.255
```

With the above configuration, as soon as BFD is reported down (3 seconds to detect), EIGRP will remove the routes installed from RIB.

The following sample output shows a summary output on hub:

```
device#show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel1, IPv4 NHRP Details
Type:Hub, NHRP Peers:2,

 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 172.17.0.1             10.0.0.1    UP 00:00:14     D
     1 172.17.0.2             10.0.0.2   BFD 00:00:03     D
```

BFD is a new state which implies that while the session is UP as seen by lower layers (IKE, IPSec and NHRP), BFD sees the session as DOWN. As usual, the state is an indication of the lower most layer where the session is not UP. Also, this applies only to the parent cache entry. This could be because it was detected as DOWN by BFD or BFD is not configured on the other side.

The following sample output shows a summary output on spoke:

```
device#show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
```

```
==========================================================================

Interface: Tunnel2, IPv4 NHRP Details
Type:Spoke, NHRP Peers:2,

 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     2 172.17.0.2             10.0.0.2   BFD 00:00:02   DT1
                              10.0.0.2    UP 00:00:02   DT2
     1 172.17.0.11           10.0.0.11    UP 00:05:35     S
```

The following sample shows output for **show ip/ipv6 nhrp** command

```
device#show ip nhrp
10.0.0.2/32 via 10.0.0.2
   Tunnel2 created 00:00:15, expire 00:04:54
   Type: dynamic, Flags: router nhop rib bfd
   NBMA address: 172.17.0.2
10.0.0.11/32 via 10.0.0.11
   Tunnel2 created 00:09:04, never expire
   Type: static, Flags: used bfd
   NBMA address: 172.17.0.11
192.168.1.0/24 via 10.0.0.1
   Tunnel2 created 00:00:05, expire 00:04:54
   Type: dynamic, Flags: router unique local
   NBMA address: 172.17.0.1
    (no-socket)
192.168.2.0/24 via 10.0.0.2
   Tunnel2 created 00:00:05, expire 00:04:54
   Type: dynamic, Flags: router rib nho
   NBMA address: 172.17.0.2
```

BFD flag here implies that there is a BFD session for this peer. This marking is only for parent entries.

The following sample shows output for **show tunnel endpoints** command

```
device#show tunnel endpoints
 Tunnel2 running in multi-GRE/IP mode

 Endpoint transport 172.17.0.2 Refcount 3 Base 0x2ABF53ED09F0 Create Time 00:00:07
   overlay 10.0.0.2 Refcount 2 Parent 0x2ABF53ED09F0 Create Time 00:00:07
   Tunnel Subblocks:
      tunnel-nhrp-sb:
         NHRP subblock has 2 entries; BFD(0x2):U
 Endpoint transport 172.17.0.11 Refcount 3 Base 0x2ABF53ED0B80 Create Time 00:09:07
   overlay 10.0.0.11 Refcount 2 Parent 0x2ABF53ED0B80 Create Time 00:09:07
   Tunnel Subblocks:
      tunnel-nhrp-sb:
         NHRP subblock has 1 entries; BFD(0x1):U
```

For every tunnel endpoint, a new text "**BFD(handle):state**" is added. State here is UP(U), DOWN(D), NONE(N) or INVALID(I).

- In case, BFD is not configured on peer or a session is not UP for the first time, then the state will be N.

The following sample shows output for **show nhrp interfaces** command. This shows the configuration (and not operational) states on the interface or globally.

```
device#show nhrp interfaces
NHRP Config State
-----------------
 Global:
    BFD: Registered

 Tunnel1:
    BFD: Disabled

 Tunnel2:
    BFD: Enabled
```

This is an internal and hidden command. This will currently display if NHRP is client of BFD and if BFD is enabled on the NHRP interface.

Example: Disabling Echo Mode When Configuring Single-Hop BFD on Unnumbered Interfaces

BFD is configured on the interface using the bfd interval command

If BFD is configured on the interface using the **bfd interval** command, BFD echo is enabled by default; this is not recommended. To disable BFD echo, configure the **no bfd echo** command under the interface.

```
interface Ethernet0/0
ip unnumbered Loopback0 poll point-to-point
ip router isis 1
bfd interval 50 min_rx 50 multiplier 3
no bfd echo
isis network point-to-point
 isis bfd
end
```

BFD is configured using the BFD template configuration

If BFD is configured using the BFD template configuration, then BFD echo is disabled by default. It is recommended that BFD echo is **not** enabled under the BFD template.

```
bfd-template single-hop max


interval min-tx 50 min-rx 50 multiplier 3
 !
interface Ethernet0/0
ip unnumbered Loopback0 poll point-to-point
bfd template max
 !
device(config)#bfd-template single-hop max
 device(config-bfd)#?
  BFD template configuration commands:
    authentication  Authentication type
    dampening       Enable session dampening
    default         Set a command to its defaults
    echo            Use echo adjunct as bfd detection mechanism.
    exit            Exit from BFD template configuration mode
    interval        Transmit interval between BFD packets
    no              Negate a command or set its defaults
```

```
device(config-bfd)#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
Configuring and monitoring BGP	"Cisco BGP Overview" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
Configuring and monitoring EIGRP	"Configuring EIGRP" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
Configuring and monitoring HSRP	"Configuring HSRP" module of the *Cisco IOS IP Application Services Configuration Guide*
Configuring and monitoring IS-IS	"Configuring Integrated IS-IS" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
Configuring and monitoring OSPF	"Configuring OSPF" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
BFD commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
EIGRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference*
IS-IS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
OSPF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
BFD IPv6 Encapsulation Support	"*BFD IPv6 Encapsulation Support*" module
OSPFv3 for BFD	"*OSPFv3 for BFD*" module

Related Topic	Document Title
Static Route Support for BFD over IPv6	"*Static Route Support for BFD over IPv6*" module

Standards and RFCs

Standard/RFC	Title
IETF Draft	*Bidirectional Forwarding Detection*, February 2009 (http://tools.ietf.org/html/draft-ietf-bfd-base-09)
IETF Draft	*BFD for IPv4 and IPv6 (Single Hop)*, February 2009 (http://tools.ietf.org/html/draft-ietf-bfd-v4v6-1hop-09

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Bidirectional Forwarding Detection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

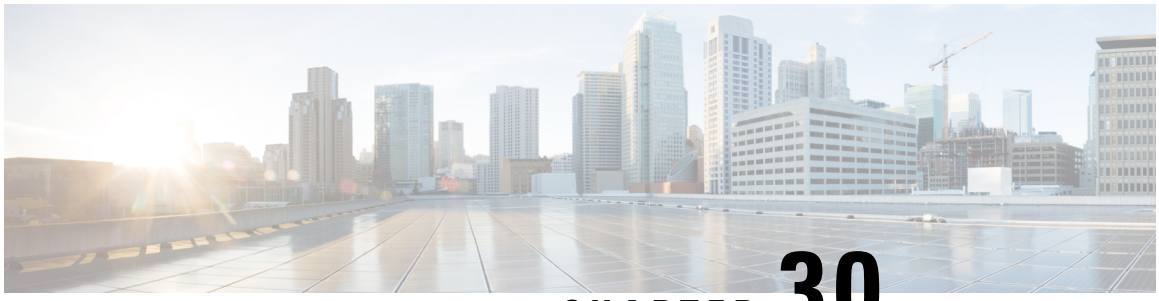
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 37: Feature Information for Bidirectional Forwarding Detection

Feature Name	Releases	Feature Information
BFD Echo Mode	12.2(33)SRB 12.4(9)T 15.0(1)S	BFD echo mode works with asynchronous BFD. Echo packets are sent by the forwarding engine and forwarded back along the same path in order to perform detection--the BFD session at the other end does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process, therefore the number of BFD control packets that are sent out between two BFD neighbors is reduced. And since the forwarding engine is testing the forwarding path on the remote (neighbor) system without involving the remote system, there is an opportunity to improve the interpacket delay variance, thereby achieving quicker failure detection times than when using BFD Version 0 with BFD control packets for the BFD session.
BFD IPv6 Encapsulation Support	Cisco IOS XE Release 3.11S	This feature extends IPv6 support for BFD. The following command was introduced or modified: **bfd interval**
BFD Multihop	15.1(3)S 15.4(1)S	This feature supports multihop BFD for IPv4 and IPv6 addresses. The following commands were introduced or modified: **authentication**, **bfd map**, **bfd-template**, **interval**, **show bfd neighbors**, **show bfd neighbor drops**.
BFD—Static Route Support	12.2(33)SRC 15.0(1)M 15.0(1)S 15.0(1)SY 15.1(2)S 15.1(1)SG 15.4(1)S	Unlike dynamic routing protocols, such as OSPF and BGP, static routing has no method of peer discovery. Therefore, when BFD is configured, the reachability of the gateway is completely dependent on the state of the BFD session to the specified neighbor. Unless the BFD session is up, the gateway for the static route is considered unreachable, and therefore the affected routes will not be installed in the appropriate RIB. A single BFD session can be used by an IPv4 static client to track the reachability of next hops through a specific interface. A BFD group can be assigned for a set of BFD-tracked static routes. The following commands were introduced or modified: **ip route static bfd** and **show ip static route bfd**.
BFD Support for IP Tunnel (GRE, with IP address)	15.1(1)SY	This feature supports BFD forwarding on point-to-point IPv4, IPv6, and GRE tunnels. The following commands were introduced or modified: **bfd** .
BFD Support over Port Channel	15.1(1)SY 15.1(2)SY	This feature supports configuring BFD timers on port channel interface. The following commands were introduced or modified: **bfd** .

Feature Name	Releases	Feature Information
BFD—VRF Support	12.2(33)SRC 15.0(1)M 15.0(1)S 15.1(1)SY	The BFD feature support is extended to be VPN Routing and Forwarding (VRF) aware to provide fast detection of routing protocol failures between provider edge (PE) and customer edge (CE) devices.
BFD—WAN Interface Support	12.2(33)SRC 15.0(1)M 15.0(1)S	The BFD feature is supported on nonbroadcast media interfaces including ATM, POS, serial, and VLAN interfaces. BFD support also extends to ATM, FR, POS, and serial subinterfaces. The **bfd interval** command must be configured on the interface to initiate BFD monitoring.
Bidirectional Forwarding Detection (standard implementation, Version 1)	12.0(31)S 12.0(32)S 12.2(33)SRB 12.2(33)SRC 12.2(18)SXE 12.2(33)SXH 12.4(9)T 12.4(11)T 12.4(15)T 15.0(1)S 15.4(1)S	This document describes how to enable the Bidirectional Forwarding Detection (BFD) protocol. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable.
HSRP Support for BFD	12.2(33)SRC 12.4(11)T 12.4(15)T	In Release 12.4(11)T, support for HSRP was added. In Release 12.2(33)SRC, the number of BFD sessions that can be created has been increased, BFD support has been extended to ATM, FR, POS, and serial subinterfaces, the BFD feature has been extended to be VRF-aware, BFD sessions are placed in an “Admin Down” state during a planned switchover, and BFD support has been extended to static routing.
IS-IS Support for BFD over IPv4	12.0(31)S 12.2(18)SXE 12.2(33)SRA 12.4(4)T 15.0(1)S 15.4(1)S	BFD support for OSPF can be configured globally on all interfaces or configured selectively on one or more interfaces. When BFD support is configured with IS-IS as a registered protocol with BFD, IS-IS receives forwarding path detection failure messages from BFD.

Feature Name	Releases	Feature Information
OSPF Support for BFD over IPv4	12.0(31)S 12.2(18)SXE 12.2(33)SRA 12.4(4)T 15.0(1)S 15.1(1)SG	BFD support for OSPF can be configured globally on all interfaces or configured selectively on one or more interfaces. When BFD support is configured with OSPF as a registered protocol with BFD, OSPF receives forwarding path detection failure messages from BFD.
SSO—BFD	12.2(33)SRE 12.2(33)SXI2 12.2(33)XNE 15.0(1)S 15.1(1)SG	Network deployments that use dual RP routers and switches have a graceful restart mechanism to protect forwarding states across a switchover. This feature enables BFD to maintain sessions in a up state across switchovers.
SSO—BFD (Admin Down)	12.2(33)SRC 15.0(1)S	To support SSO, BFD sessions are placed in an “Admin Down” state during a planned switchover. The BFD configuration is synched from the active to standby processor, and all BFD clients re-register with the BFD process on the standby processor.
sVTI Support on BFD	Cisco IOS-XE 17.6.4 Cisco IOS-XE 17.9.1a	sVTI support on BFD is introduced in Cisco IOS-XE 17.6.4 release. This feature is also supported in Cisco IOS-XE 17.9.1a release.



CHAPTER 30

Static Route Support for BFD over IPv6

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Static Route Support for BFD over IPv6

Using the BFDv6 protocol to reach the static route next hop ensures that an IPv6 static route is inserted only in the IPv6 Routing Information Base (RIB) when the next-hop neighbor is reachable. Using the BFDv6 protocol also can remove the IPv6 static route from the IPv6 RIB when the next hop becomes unreachable.

A user can configure IPv6 static BFDv6 neighbors. These neighbor can operate in one of two modes: associated (which is the default) and unassociated. A neighbor can be transitioned between the two modes without interrupting the BFDv6 session associated with the neighbor.

BFDv6 Associated Mode

In Bidirectional Forwarding Detection for IPv6 (BFDv6) associated mode, an IPv6 static route is automatically associated with an IPv6 static BFDv6 neighbor if the static route next hop exactly matches the static BFDv6 neighbor.

An IPv6 static route requests a BFDv6 session for each static BFDv6 neighbor that has one or more associated IPv6 static routes and is configured over an interface on which BFD has been configured. The state of the

BFDv6 session will be used to determine whether the associated IPv6 static routes are inserted in the IPv6 RIB. For example, static routes are inserted in the IPv6 RIB only if the BFDv6 neighbor is reachable, and the static route is removed from the IPv6 RIB if the BFDv6 neighbor subsequently becomes unreachable.

BFDv6 associated mode requires you to configure a BFD neighbor and static route on both the device on which the BFD-monitored static route is required and on the neighboring device.

BFDv6 Unassociated Mode

An IPv6 static BFD neighbor may be configured as unassociated. In this mode, the neighbor is not associated with static routes, and the neighbor always requests a BFDv6 session if the interface has been configured for BFDv6.

Unassociated mode is useful in the following situations:

- Bringing up a BFDv6 session in the absence of an IPv6 static route—This case occurs when a static route is on router A, with router B as the next hop. Associated mode requires you to create both a static BFD neighbor and static route on both routers in order to bring up the BFDv6 session from B to A. Specifying the static BFD neighbor in unassociated mode on router B avoids the need to configure an unwanted static route.
- Transition to BFD monitoring of a static route—This case occurs when existing IPv6 static routes are inserted in the IPv6 RIB. Here, you want to enable BFD monitoring for these static routes without any interruption to traffic. If you configure an attached IPv6 static BFD neighbor, then the static routes will immediately be associated with the new static BFD neighbor. However, because a static BFD neighbor starts in a down state, the associated static routes are then removed from the IPv6 RIB and are reinserted when the BFDv6 session comes up. Therefore, you will see an interruption in traffic. This interruption can be avoided by configuring the static BFD neighbor as unassociated, waiting until the BFDv6 session has come up, and then reconfiguring the static BFD neighbor as associated.
- Transition from BFD monitoring of a static route—In this case, IPv6 static routes are monitored by BFD and inserted in the RIB. Here, you want to disable BFD monitoring of the static routes without interrupting traffic flow. This scenario can be achieved by first reconfiguring the static BFD neighbor as detached (thus disassociating the neighbor from the static routes) and then deconfiguring the static BFD neighbor.

How to Configure Bidirectional Forwarding Detection for IPv6

Specifying a Static BFDv6 Neighbor

An IPv6 static BFDv6 neighbor is specified separately from an IPv6 static route. An IPv6 static BFDv6 neighbor must be fully configured with the interface and neighbor address and must be directly attached to the local router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]

Example:

```
Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1
```

Specifies static route IPv6 BFDv6 neighbors.

Associating an IPv6 Static Route with a BFDv6 Neighbor

IPv6 static routes are automatically associated with a static BFDv6 neighbor. A static neighbor is associated with a BFDv6 neighbor if the static next-hop explicitly matches the BFDv6 neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]
4. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ipv6 route static bfd** [**vrf** *vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]

Example:

```
Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1
```

Specifies static route BFDv6 neighbors.

Step 4 **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length* {*ipv6-address* | *interface-type interface-number ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

Example:

```
Device(config)# ipv6 route 2001:DB8::/64 gigabitethernet 0/0/0 2001::1
```

Establishes static IPv6 routes.

Configuration Examples for Static Route Support for BFD over IPv6

Example: Specifying an IPv6 Static BFDv6 Neighbor

The following example specifies a fully configured IPv6 static BFDv6 neighbor. The interface is GigabitEthernet 0/0/0 and the neighbor address is 2001::1.

```
Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1
```

Example: Associating an IPv6 Static Route with a BFDv6 Neighbor

In this example, the IPv6 static route 2001:DB8::/32 is associated with the BFDv6 neighbor 2001::1 over the GigabitEthernet 0/0/0 interface:

```
Device(config)# ipv6 route static bfd gigabitethernet 0/0/0 2001::1
Device(config)# ipv6 route 2001:DB8::/32 gigabitethernet 0/0/0 2001::1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping
Static Route Support for BFD over IPv6	" *Bidirectional Forwarding Detection*" module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

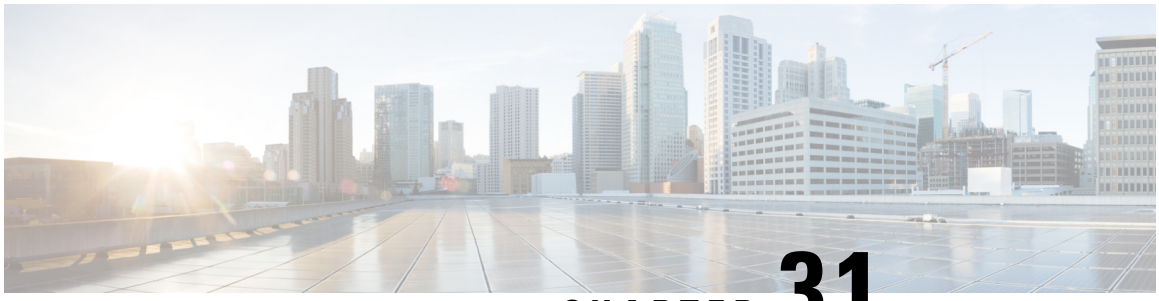
Feature Information for Static Route Support for BFD over IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 38: Feature Information for Static Route Support for BFD over IPv6

Feature Name	Releases	Feature Information
Static Route Support for BFD over IPv6	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.6S	Using the BFDv6 protocol to reach the static route next hop ensures that an IPv6 static route is inserted only in the IPv6 Routing Information Base (RIB) when the next-hop neighbor is reachable. Using the BFDv6 protocol also can remove the IPv6 static route from the IPv6 RIB when the next hop becomes unreachable. The following commands were introduced or modified: **debug bfd**, **debug ipv6 static**, **ipv6 static**, **ipv6 route static bfd**, **monitor event ipv6 static**, **show ipv6 static**.



CHAPTER 31

OSPFv3 for BFD

The Bidirectional Forwarding Detection protocol supports OSPFv3.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About OSPFv3 for BFD

The Bidirectional Forwarding Detection (BFD) protocol supports Open Shortest Path First version 3 (OSPFv3).

How to Configure OSPFv3 for BFD

Configuring BFD Support for OSPFv3

This section describes the procedures for configuring BFD support for OSPFv3, so that OSPFv3 is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. You can either configure BFD support for OSPFv3 globally on all interfaces or configure it selectively on one or more interfaces.

There are two methods for enabling BFD support for OSPFv3:

- You can enable BFD for all of the interfaces for which OSPFv3 is routing by using the **bfd all-interfaces** command in router configuration mode. You can disable BFD support on individual interfaces using the **ipv6 ospf bfd disable** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which OSPFv3 is routing by using the **ipv6 ospf bfd** command in interface configuration mode.

Note OSPF will only initiate BFD sessions for OSPF neighbors that are in the FULL state.

Configuring Baseline BFD Session Parameters on the Interface

Repeat this task for each interface over which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Device(config-if)# bfd interval 50 min_rx 50 multiplier 5`	Enables BFD on the interface.

Configuring BFD Support for OSPFv3 for All Interfaces

Before you begin

OSPFv3 must be running on all participating devices. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id* [**vrf** *vpn-name*]
4. **bfd all-interfaces** [**strict-mode**]
5. **exit**
6. **show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address* | **ipv6** *ipv6-address*] [**details**]
7. **show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# ipv6 router ospf 2`	Configures an OSPFv3 routing process.
Step 4	**bfd all-interfaces** [**strict-mode**] **Example:** `Device(config-router)# bfd all-interfaces`	Enables BFD for all interfaces participating in the routing process. **[strict-mode]** - BFD session is established in the strict-mode. In the strict-mode, the OSPF session is not established till the BFD session is established.
Step 5	**exit** **Example:** `Device(config-router)# exit`	Enter this command twice to go to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address* | **ipv6** *ipv6-address*] [**details**] **Example:** `Device# show bfd neighbors detail`	(Optional) Displays a line-by-line listing of existing BFD adjacencies.
Step 7	**show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**] **Example:** `Device# show ipv6 ospf`	(Optional) Displays general information about OSPFv3 routing processes. If BFD is enabled in strict-mode, the command output displays `BFD is enabled in strict mode`.

Configuring BFDv6 Support for OSPFv3 on One or More OSPFv3 Interfaces

Before you begin

OSPFv3 must be running on all participating devices. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 ospf bfd** [**disable**] [**strict-mode**]
5. **exit**
6. **show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address*| **ipv6** *ipv6-address*] [**details**]
7. **show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies an interface type and number, and places the device in interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface GigabitEthernet 0/0/0`	
Step 4	**ipv6 ospf bfd** [**disable**] [**strict-mode**] **Example:** `Device(config-if)# ipv6 ospf bfd`	Enables BFD on a per-interface basis for one or more interfaces associated with the OSPFv3 routing process. **[strict-mode]** - BFD session is established in the strict-mode. In the strict-mode, the OSPF session is not established till the BFD session is established.
Step 5	**exit** **Example:** `Device(config-router)# exit`	Enter this command twice to go to privileged EXEC mode.
Step 6	**show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address*| **ipv6** *ipv6-address*] [**details**] **Example:** `Device# show bfd neighbors detail`	(Optional) Displays a line-by-line listing of existing BFD adjacencies.
Step 7	**show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**] **Example:** `Device# show ipv6 ospf`	(Optional) Displays general information about OSPFv3 routing processes. If BFD is enabled in strict-mode, the command output displays `BFD is enabled in strict mode`.

Retrieving BFDv6 Information for Monitoring and Troubleshooting

SUMMARY STEPS

1. **enable**
2. **monitor event ipv6 static** [**enable** | **disable**]
3. **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix-length*] [**interface** *type number* | **recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail**
4. **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix-length*] [**interface** *type number* | **recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail**]
5. **debug ipv6 static**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**monitor event ipv6 static** [**enable** | **disable**] **Example:** `Device# monitor event ipv6 static enable`	Enables the use of event trace to monitor the operation of the IPv6 static and IPv6 static BFDv6 neighbors.
Step 3	**show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix-length*] [**interface** *type number* | **recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail** **Example:** `Device# show ipv6 static vrf vrf1 detail`	Displays the BFDv6 status for a static route associated with a static BFDv6 neighbor.
Step 4	**show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix-length*] [**interface** *type number* | **recursive**] [**vrf** *vrf-name*] [**bfd**] [**detail**] **Example:** `Device# show ipv6 static vrf vrf1 bfd`	Displays static BFDv6 neighbors and associated static routes.
Step 5	**debug ipv6 static** **Example:** `Device# debug ipv6 static`	Enables BFDv6 debugging.

Configuration Examples for OSPFv3 for BFD

Example: Displaying OSPF Interface Information about BFD

The following display shows that the OSPF interface is enabled for BFD:

```
Device# show ipv6 ospf interface

Serial10/0 is up, line protocol is up
  Link Local Address FE80::A8BB:CCFF:FE00:6500, Interface ID 42
  Area 1, Process ID 1, Instance ID 0, Router ID 10.0.0.1
  Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT, BFD enabled
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:07
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.0.1
  Suppress hello for 0 neighbor(s)
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
OSPFv3 for BFD	"*Bidirectional Forwarding Detection*" module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

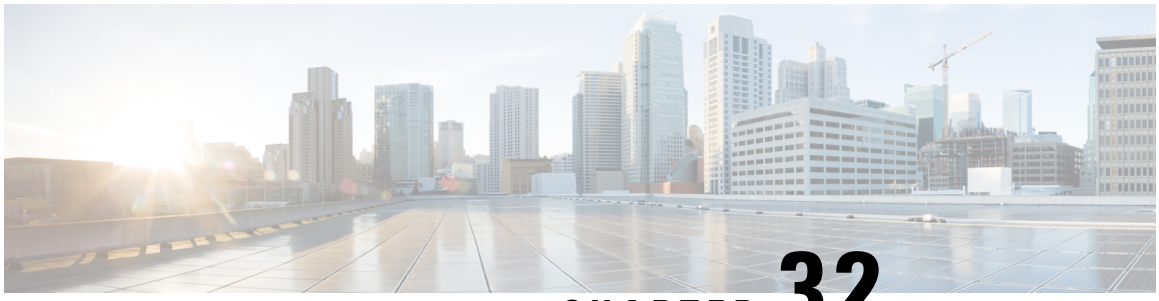
Feature Information for OSPFv3 for BFD

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 39: Feature Information for OSPFv3 for BFD

Feature Name	Releases	Feature Information
OSPFv3 for BFD	Cisco IOS XE Release 2.1	BFD supports the dynamic routing protocol OSPFv3. The following commands were introduced or modified: **bfd**, **bfd all-interfaces**, **debug bfd**, **ipv6 router ospf**, **show bfd neighbors**, **show ipv6 ospf**, **show ipv6 ospf interface**, **show ospfv3**, **show ospfv3 interface**.



CHAPTER 32

BFD on BDI Interfaces

The Cisco BFD on BDI Interfaces feature alleviates limitations on the maximum number of interfaces per system that switched virtual interfaces (SVI) impose. This document describes how to configure the Bidirectional Forwarding Detection (BFD) protocol on bridge domain interfaces (BDIs).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About BFD on Bridge Domain Interfaces

BFD on Bridge Domain Interfaces

Each BDI is associated with a bridge domain on which traffic is mapped using criteria defined and configured on the associated Ethernet flow points (EFPs). You can associate either single or multiple EFPs with a given bridge domain. Thus you can establish a BFD single-hop session over BDI interfaces that are defined in either a global table or a VPN routing and forwarding (VRF) table, and all existing single-hop BFD clients will be supported for BFD over BDI.

The Cisco BFD on BDI feature does not affect BFD stateful switchover (SSO) on platforms that are SSO capable.

How to Configure BFD on BDI Interfaces

Enabling BFD on a Bridge Domain Interface

Perform these steps to enable single hop BFD on an individual BDI interface.

Note Multihop BFD is not interface specific so you do not need BDI interface-level configuration to establish multihop BFD sessions.

Before you begin

Two or more nodes must be connected.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **exit**

DETAILED STEPS

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Router(config)# interface bdi 100
```

Configures a bridge domain interface and enters interface configuration mode.

Step 4 **ip address** *ip-address mask*

Example:

```
Router(config-if)# ip address 10.201.201.1 255.255.255.0
```

Configures an IP address for the interface.

Step 5 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode and returns to global configuration mode.

Associating an Ethernet Flow Point with a Bridge Domain

Before you begin

BFD must be enabled on both nodes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot/port*
4. **no ip address**
5. **negotiation auto**
6. **cdp enable**
7. **service instance** *id service-type*
8. **encapsulation dot1q** *vlan-id*
9. **rewrite ingress tag pop 1 symmetric**
10. **exit**
11. **exit**
12. **bridge-domain** *vlan-id*

DETAILED STEPS

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type slot/subslot/port*

Example:

```
Router(config)# interface GigabitEthernet0/0/3
```

Configures an interface type and enters interface configuration mode.

Step 4 **no ip address**

Example:

```
Router(config-if)# no ip address
```

Disables IP processing.

Step 5 **negotiation auto**

Example:

```
Router(config-if)# negotiation auto
```

Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the interface.

Step 6 **cdp enable**

Example:

```
Router(config-if)# cdp enable
```

Enables Cisco Discovery Protocol on the interface.

Step 7 **service instance** *id service-type*

Example:

```
Router(config-if)# service instance 2 ethernet
```

Configures an Ethernet service instance and enters service instance configuration mode.

Step 8 **encapsulation dot1q** *vlan-id*

Example:

```
Router(config-if-srv)# encapsulation dot1q 2
```

Enables IEEE 802.1Q encapsulation of traffic on the subinterface.

Step 9 **rewrite ingress tag pop 1 symmetric**

Example:

```
Router(config-if-srv)# rewrite ingress tag pop 1 symmetric
```

Specifies removal of the outermost tag from the frame ingressing the service instance and the addition of a tag in the egress direction.

Step 10 **exit**

Example:

```
Router(config-if)# exit
```

Exits service instance configuration mode and returns to interface configuration mode.

Step 11 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode and returns to global configuration mode.

Step 12 **bridge-domain** *vlan-id*

Example:

```
Router(config)# bridge-domain 2
```

Associates the bridge domain with the Ethernet flow point.

Example:

What to do next

Configuration Examples for BFD on BDI Interfaces

Examples for BFD on BDI Interfaces

The following example shows how to configure BFD on a BDI.

```
Router#show bfd neighbors

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
10.1.1.2                              2049/1         Up        Up        BD2
Router#
Router#show running interface gi0/0/3
Building configuration...

Current configuration : 230 bytes
!
interface GigabitEthernet0/0/3
no ip address
ip pim passive
ip igmp version 3
negotiation auto
cdp enable
service instance 2 ethernet
  encapsulation dot1q 2
```

```
  rewrite ingress tag pop 1 symmetric
  bridge-domain 2
!
end

Router#show running interface bdi2

Building configuration...

Current configuration : 127 bytes
!
interface BDI2
ip address 10.1.1.3 255.255.255.0
bfd interval 100 min_rx 100 multiplier 3
bfd neighbor ipv4 10.1.1.2
end
```

And similarly for the other node:

```
Router2#show running interface bdi2

Building configuration...

Current configuration : 127 bytes
!
interface BDI2
ip address 10.1.1.2 255.255.255.0
bfd interval 100 min_rx 100 multiplier 3
bfd neighbor ipv4 10.1.1.3
end

ED3#show run int gig0/0/3
Building configuration...

Current configuration : 195 bytes
!
interface GigabitEthernet0/0/3
no ip address
negotiation auto
cdp enable
service instance 2 ethernet
  encapsulation dot1q 2
  rewrite ingress tag pop 1 symmetric
  bridge-domain 2
!
end

Router2#show bfd neighbors

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
10.1.1.3                                1/2049       Up        Up        BD2
ED3#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
Configuring and monitoring BGP	"Cisco BGP Overview" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
Configuring and monitoring EIGRP	"Configuring EIGRP" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
Configuring and monitoring HSRP	"Configuring HSRP" module of the *Cisco IOS IP Application Services Configuration Guide*
Configuring and monitoring IS-IS	"Configuring Integrated IS-IS" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
Configuring and monitoring OSPF	"Configuring OSPF" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
BFD commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
EIGRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Application Services Command Reference*
IS-IS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
OSPF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
BFD IPv6 Encapsulation Support	"*BFD IPv6 Encapsulation Support*" module
OSPFv3 for BFD	"*OSPFv3 for BFD*" module
Static Route Support for BFD over IPv6	"*Static Route Support for BFD over IPv6*" module

Standards and RFCs

Standard/RFC	Title
IETF Draft	*Bidirectional Forwarding Detection*, February 2009 (http://tools.ietf.org/html/draft-ietf-bfd-base-09)
IETF Draft	*BFD for IPv4 and IPv6 (Single Hop)*, February 2009 (http://tools.ietf.org/html/draft-ietf-bfd-v4v6-1hop-09

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

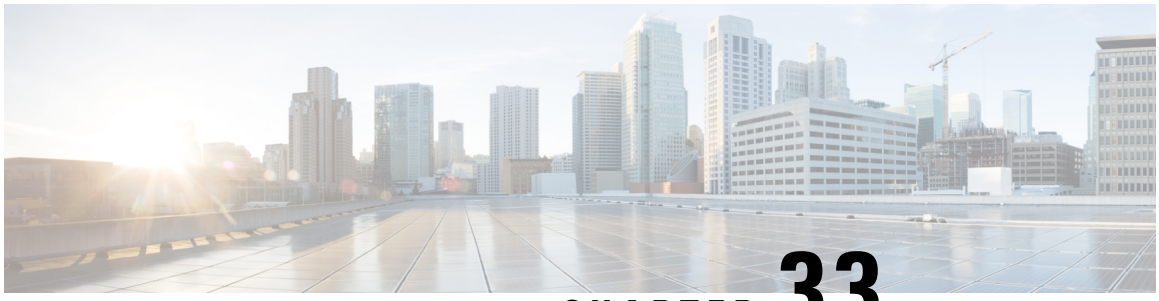
Feature Information for BFD on Bridge Domain Interfaces

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 40: Feature Information for BFD on Bridge Domain Interfaces

Feature Name	Releases	Feature Information
BFD on Bridge Domain Interfaces	Cisco IOS XE Release 3.5S	This feature supports BFD on Bridge Domain Interfaces.



CHAPTER 33

BFD Single-Hop Authentication

The BFD Single-Hop Authentication feature enables authentication for single-hop Bidirectional Forwarding Detection (BFD) sessions between two directly connected devices. This feature supports Message Digest 5 (MD5) and Secure Hash Algorithm 1 (SHA-1) authentication types.

This module explains the BFD Single-Hop Authentication feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for BFD Single-Hop Authentication

You must configure keys and key chains on both connected devices that are involved in a BFD session. You must configure the algorithm and the key chain on both devices in such a way that the configurations match.

Restrictions for BFD Single-Hop Authentication

- If key chains are removed from the established BFD single-hop sessions or no active keys are present in the key chain, the BFD template and the map entry are invalidated. Such invalidation is considered as a map entry deletion.
- Meticulous keyed MD5 authentication and meticulous keyed SHA-1 are not supported in In-Service Software Upgrade (ISSU) because checkpointing of sequence numbers does not occur in all packets.
- Meticulous MD5 and meticulous SHA-1 authentication types are not preserved after Route Processor (RP) failures in Stateful Switchover (SSO) mode. The sessions could flap causing link instability of the registered protocols.
- Only timers with values greater than or equal to 50 milliseconds are supported.
- The authentication type negotiation and key exchange between two BFD peers does not occur.
- When there is a missing key chain or when keys are not configured in a key chain, the BFD template and its associated map entries are invalidated, and the BFD session is not created.
- You can apply Bidirectional Forwarding Detection (BFD) single-hop Authentication in a BFD-template configuration only. You cannot apply BFD single-hop authentication in legacy configurations.

Information About BFD Single-Hop Authentication

Benefits of BFD Single-Hop Authentication

Using the Message Digest 5 (MD5) and Secure Hash Algorithm 1 (SHA-1) authentication methods defined in RFC 5880, the BFD Single Hop Authentication feature provides security against attacks on data links between a pair of directly connected devices involved in a BFD session. This feature is applied on data links between a BFD source-destination pair that communicates through IPv4 and IPv6 protocols across a single IP hop that is associated with an incoming interface. The communication may occur through physical media, virtual circuits, and tunnels.

Role of BFD Single-Hop Authentication in Preventing Denial of Service Attacks

To prevent denial of service (DoS) attacks, a BFD single-hop session validates the sequence number of a packet on receiving the packet. Detect multiplier is the number of missing BFD hello messages from another BFD device before the local device detects a fault in the forwarding path. The detect multiplier is used to determine the detect timer. The following are the ranges of valid sequence numbers that are accepted by the BFD Single-Hop Authentication feature:

- For nonmeticulous keyed types: Last received sequence number to (last received sequence number + 3 * detect multiplier)
- For meticulous keyed types: Last received sequence number + 1) to (last received sequence number + 3 * detect multiplier)

Note For BFD, (transmit interval) * (detect multiplier) = detect timer. If a BFD control packet is not received from the remote system within the detect-timer interval, a failure has occurred.

How to Configure BFD Single-Hop Authentication

Configuring Key Chains

Perform this task on one of the two devices that are involved in a BFD session, and repeat the steps on the other device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *chain-name*
4. **key** *key-id*
5. **key-string** *text*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *chain-name* **Example:** `Device(config)# key chain chain1`	Defines an authentication key chain needed to enable authentication for routing protocols and enters key-chain configuration mode.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Defines an authentication key on the key chain and enters keychain-key configuration mode.
Step 5	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string key1`	Defines an authentication string for a key.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-keychain-key)# end`	Exits keychain-key configuration mode and returns to privileged EXEC mode.

Configuring a BFD Template with Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bfd-template single-hop** *template-name*
4. **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*
5. **authentication** *authentication-type* **keychain** *keychain-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bfd-template single-hop** *template-name* **Example:** `Device(config)# bfd-template single-hop template1`	Creates a BFD template and enters BFD configuration mode.
Step 4	**interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value* **Example:** `Device(config-bfd)# interval min-tx 120 min-rx 100 multiplier 3`	Configures transmit and receive intervals between BFD packets and specifies the number of consecutive BFD control packets that must be missed before BFD declares that a peer is unavailable.
Step 5	**authentication** *authentication-type* **keychain** *keychain-name* **Example:** `Device(config-bfd)# authentication sha-1 keychain keychain1`	Configures authentication in a BFD template for single-hop sessions.
Step 6	**end** **Example:**	Exits BFD configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-bfd)# end`	

Configuring a Single-Hop Template on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bfd template** *template-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Enters interface configuration mode.
Step 4	**bfd template** *template-name* **Example:** `Device(config-if)# bfd template bfdtemplate`	Binds a single-hop BFD template to an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying BFD Single-Hop Authentication

SUMMARY STEPS

1. **show bfd drops**
2. **show bfd neighbor**

DETAILED STEPS

Step 1 **show bfd drops**

Example:

```
Device> show bfd drops
```

This command displays the number of dropped packets in BFD.

Step 2 **show bfd neighbor**

Example:

```
Device> show bfd neighbor
```

This command displays a line-by-line listing of existing BFD adjacencies.

Configuration Examples for BFD Single-Hop Authentication

Example: Configuring Key Chains

```
Device> enable
Device# configure terminal
Device(config)# key chain chain1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string key1
Device(config-keychain-key)# end
```

Example: Configuring a BFD Template with Authentication

```
Device> enable
Device# configure terminal
Device(config)# bfd-template single-hop template1
Device(bfd-config)# interval min-tx 120 min-rx 100 multiplier 3
Device(bfd-config)# authentication sha-1 keychain keychain1
Device(bfd-config)# end
```

Example: Configuring a Single-Hop Template on an Interface

```
Device> enable
Device# configure terminal
Device(config)# key chain chain1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string key1
Device(config-keychain-key)# end
```

Example: Verifying BFD Single-Hop Authentication

Sample Output for the show bfd neighbor command

```
Device> show bfd neighbor

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
192.168.0.2                            1/12          Up        Up        Et0/0
Session state is UP and using echo function with 300 ms interval.
Session Host: Software
OurAddr: 192.168.0.1
Handle: 12
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holddown (hits): 0(0), Hello (hits): 1000(62244)
Rx Count: 62284, Rx Interval (ms) min/max/avg: 1/2436/878 last: 239 ms ago
Tx Count: 62247, Tx Interval (ms) min/max/avg: 1/1545/880 last: 246 ms ago
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: Stub CEF
Template: my-template
Authentication(Type/Keychain): sha-1/my-chain
Uptime: 00:22:06
Last packet: Version: 1                - Diagnostic: 0
             State bit: Up             - Demand bit: 0
             Poll bit: 0               - Final bit: 0
             Multiplier: 3             - Length: 24
             My Discr.: 12             - Your Discr.: 1
             Min tx interval: 1000000  - Min rx interval: 1000000
             Min Echo interval: 300000
```

Sample Output for the show bfd drops command.

```
Device> show  bfd drops

BFD Drop Statistics
                        IPV4    IPV6    IPV4-M  IPV6-M  MPLS_PW MPLS_TP_LSP
Invalid TTL             0       0       0       0       0       0
BFD Not Configured      0       0       0       0       0       0
No BFD Adjacency        0       0       0       0       0       0
Invalid Header Bits     0       0       0       0       0       0
Invalid Discriminator   0       0       0       0       0       0
Session AdminDown       0       0       0       0       0       0
Authen invalid BFD ver  0       0       0       0       0       0
Authen invalid len      0       0       0       0       0       0
Authen invalid seq      0       0       0       0       0       0
Authen failed           0       0       0       0       0       0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
IP Routing: Protocol-Independent Commands	*Cisco IOS IP Routing Protocol-Independent Command Reference*

Standards and RFCs

Standard/RFC	Title
RFC 5880	*Bidirectional Forwarding Detection*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

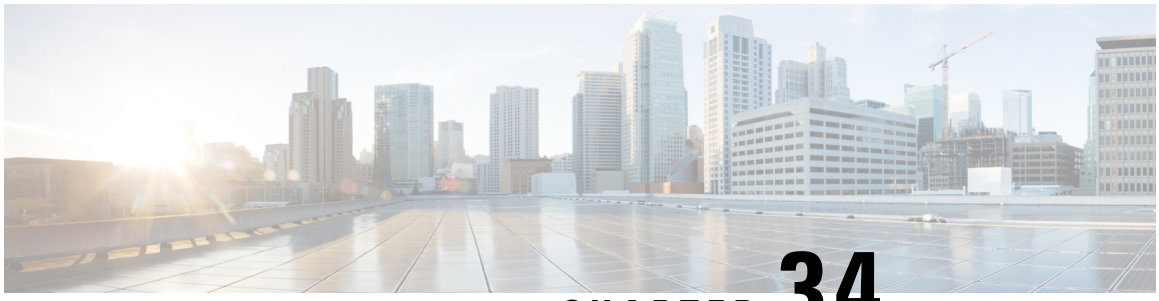
Feature Information for BFD Single-Hop Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 41: Feature Information for BFD Single Hop Authentication

Feature Name	Releases	Feature Information
BFD Single-Hop Authentication	15.2(4)S	The BFD Single-Hop Authentication feature enables authentication for single hop BFD sessions between directly connected devices. This feature supports Message Digest 5 (MD5) and Secure Hash Algorithm 1 (SHA1) authentication types. The following commands were introduced or modified: **authentication (BFD)**, **bfd template**, **bfd-template**, **show bfd drops** and **show bfd neighbors**.



CHAPTER 34

BFD Multihop Support for IPv4 Static Routes

The BFD Multihop Support for IPv4 Static Routes feature enables detection of IPv4 network failure between paths that are not directly connected. If a Bidirectional Forwarding Detection (BFD) session is up (that is, the next-hop destination is reachable), IPv4 static routes that are associated with IPv4 static BFD configuration are added to a routing table. If the BFD session is down, the routing table removes all associated static routes from the routing table.

This feature is applicable on different kinds of interfaces such as physical, subinterface, and virtual tunnels and across intra-area and interarea topologies.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for BFD Multihop Support for IPv4 Static Routes

- The BFD destination for which an IPv4 static route has to be configured must be reachable by all devices.
- The configured device must have at least one static route with the next-hop destination as a BFD destination for an associated session. If not, the BFD session is not created on the device.

Information About BFD Multihop Support for IPv4 Static Routes

BFDv4 Associated Mode

In Bidirectional Forwarding Detection for IPv4 (BFDv4) associated mode, an IPv4 static route is automatically associated with an IPv4 static BFDv4 multihop destination address if the static route next hop exactly matches the static BFDv4 multihop destination address.

The state of the BFDv4 session is used to determine whether the associated IPv4 static routes are added in the IPv4 routing information base (RIB). For example, static routes are added in the IPv4 RIB only if the BFDv4 multihop destination is reachable, and the static routes are removed from the IPv4 RIB if the BFDv4 multihop destination subsequently becomes unreachable.

BFDv4 Unassociated Mode

In Bidirectional Forwarding Detection for IPv4 (BFDv4), an IPv4 static BFD multihop destination can be configured in unassociated mode. In unassociate mode, a BFD neighbor is not associated with a static route, and the BFD sessions are requested if the IPv4 static BFD is configured.

Unassociated mode is useful in the following scenario:

- Absence of an IPv4 static route—This scenario occurs when a static route is on device A, and device B is the next hop. In associated mode, you must create both a static BFD multihop destination address and a static route on both devices to bring up the BFDv4 session from device B to device A. Specifying the static BFD multihop destination in unassociated mode on device B avoids the need to configure an unwanted static route.

How to Configure BFD Multihop Support for IPv4 Static Routes

Configuring BFD Multihop IPv4 Static Routes

Before you begin

- Specify a BFD destination address which is same as the IPv4 static route next hop or gateway address.
- Configure a BFD map and a BFD multihop template for an interface on the device. The destination address and source address configured for a BFD map must match the BFD static multihop configuration and the source address must be a valid IP address configured for an interface in the routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask ip-address*
4. **ip route static bfd** *multihop-destination-address multihop-source-address*
5. **ip route static bfd** *multihop-destination-address multihop-source-address* **unassociate**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask ip-address* **Example:** `Device(config)# ip route 192.0.2.0 255.255.255.0 10.1.1.2`	Configures an IPv4 static route that BFD multihop uses to monitor static routes.
Step 4	**ip route static bfd** *multihop-destination-address multihop-source-address* **Example:** `Device(config)# ip route static bfd 192.0.2.1 10.1.1.1`	Configures the static IPv4 BFD multihop to be associated with a static IPv4 route.
Step 5	**ip route static bfd** *multihop-destination-address multihop-source-address* **unassociate** **Example:** `Device(config)# ip route static bfd 192.0.2.1 10.1.1.1 unassociate`	(Optional) Configures the static IPv4 BFD multihop to be associated with a static IPv4 route in unassociated mode.
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Verifying BFD Multihop Support for IPv4 Static Routes

The following show commands can be used to verify IPv4 static routes for BFD multihop:

SUMMARY STEPS

1. **show bfd neighbor**
2. **show ip static route bfd**

DETAILED STEPS

Step 1 **show bfd neighbor**

Displays a line-by-line listing of existing BFD adjacencies.

Step 2 **show ip static route bfd**

Displays information about the IPv4 static BFD configured parameters.

Configuration Examples for BFD Multihop Support for IPv4 Static Routes

Example: Configuring BFD Multihop for IPv4 Static Routes in Associated Mode

```
Device> enable
Device# configure terminal
Device(config)# bfd map ipv4 192.0.2.1/32  10.1.1.1/32 test
Device(config)# bfd-template multi-hop test
Device(config-bfd)# interval min-tx 51 min-rx 51 multiplier 3
Device(config-bfd)# exit
Device(config)# ip route 192.0.2.0 255.255.255.0 10.1.1.2
Device(config)# interface GigabitEthernet 1/1
Device(config-if)# ip address 10.1.1.1 255.255.0.0
Device(config-if)# exit
Device(config)# ip route static bfd 192.0.2.1 10.1.1.1
Device(config)# end
```

Example: Configuring IPv4 Static Multihop for BFD in Unassociated Mode

```
Device> enable
Device# configure terminal
Device(config)# bfd map ipv4 192.0.2.1/32  10.1.1.1/32 test
Device(config)# bfd-template multi-hop test
Device(config-bfd)# interval min-tx 51 min-rx 51 multiplier 3
Device(config-bfd)# exit
Device(config)# ip route 192.0.2.0 255.255.255.0 10.1.1.2
Device(config)# interface GigabitEthernet 1/1
Device(config-if)# ip address 10.1.1.1 255.255.0.0
Device(config-if)# exit
Device(config)# ip route static bfd 192.0.2.1 10.1.1.1 unassociate
Device(config)# end
```

Additional References for BFD Multihop Support for IPv4 Static Routes

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
IP Routing: Protocol Independent commands	*IP Routing Protocol-Independent Command Reference*

Standards and RFCs

Standard/RFC	Title
RFC 5883	*BFD for Multihop Paths*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

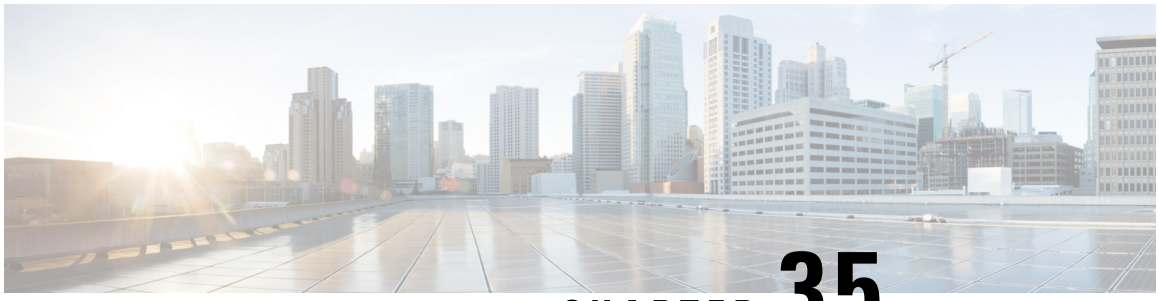
Feature Information for BFD Multihop Support for IPv4 Static Routes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 42: Feature Information for BFD Multihop Support for IPv4 Static Routes

Feature Name	Releases	Feature Information
BFD Multihop Support for IPv4 Static Routes	Cisco IOS XE Release 3.9S	The BFD Multihop Support for IPv4 Static Routes feature enables detection of IPv4 network failure between paths that are not directly connected. If a Bidirectional Forwarding Detection (BFD) session is up (that is, the next-hop destination is reachable), IPv4 static routes that are associated with IPv4 static BFD configuration are added to a routing table. If the BFD session is down, the routing table removes all associated static routes from the routing table. The following commands were modified: **ip route static bfd** and **show ip static route bfd**.



CHAPTER 35

IS-IS IPv6 Client for BFD

When Bidirectional Forwarding Detection (BFD) support is configured with Intermediate System To Intermediate System (IS-IS) as a registered protocol with BFD, IS-IS receives forwarding path detection failure messages from BFD.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for IS-IS IPv6 Client for BFD

- IS-IS must be running on all participating devices.
- The baseline parameters for BFD sessions must be configured on the interfaces that run BFD sessions to BFD neighbors.

Information About IS-IS IPv6 Client for BFD

IS-IS BFD Topology

When BFD support is configured with IS-IS as a registered protocol with BFD, IS-IS receives forwarding path detection failure messages from BFD. BFD support for IS-IS can be configured in either router address-family configuration mode or interface configuration mode. IS-IS IPv6 can run in single-topology or in Multi-Topology (MT) mode.

IS-IS BFD supports both IPv4 and IPv6 on the same adjacency for single-topology or multi-topology mode. If BFD is enabled for both IPv4 and IPv6, IS-IS sends two BFD session creation requests to BFD. For single-topology mode, the IS-IS adjacency state can only be UP if both BFD sessions are UP. If either of the BFD sessions is DOWN, the associated IS-IS adjacency state is also DOWN. For MT mode, the IS-IS adjacency state can be UP as long as one of topologies has a BFD session in an UP state.

IS-IS BFD IPv6 Session Creation

IS-IS requests a BFD session for the interface and IPv6 address of the neighboring device when all of the following conditions are met:

- An IS-IS adjacency entry exists.
- The Address Family Identifier (AFI) specific peer interface address is known.
- IS-IS BFD is enabled for that AFI on an interface.
- IS-IS is enabled for that AFI on the local interface.
- If the neighboring device supports RFC 6213, BFD must be enabled for the specified Multi-Topology Identifier (MTID) or Network Layer Protocol Identifier (NLPID).

IS-IS BFD IPv6 Session Deletion

When IS-IS BFD IPv6 is disabled on an interface, IS-IS removes related BFD sessions for IPv6 from the adjacent device. When the IS-IS adjacency entry is deleted, all BFD sessions are also deleted. IS-IS requests BFD to remove each BFD session that it has requested when any of the following events occur:

- The IS-IS instance is deleted or un-configured.
- The IS-IS adjacency entry is deleted.
- IS-IS BFD is disabled on the next hop interface for an address-family.
- The neighboring device supports RFC 6213 and indicates that it no longer supports BFD for the specified MTID or NLPID.

How to Configure ISIS IPv6 Client for BFD

Configuring IS-IS IPv6 Client Support for BFD on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** *ipv6-address/mask*
5. **isis ipv6 bfd**
6. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet 6/0/0
```

Enters interface configuration mode.

Step 4 **ipv6 address** *ipv6-address/mask*

Example:

```
Device(config-if)# ipv6 address 19:1:1::4/64
```

Configures IPv6.

Step 5 **isis ipv6 bfd**

Example:

```
Device(config-if)# isis ipv6 bfd
```

Enables IPv6 BFD on a specific interface that is configured for IS-IS.

Step 6 **end**

Example:

```
Device(config-if)# end
```

Exits interface configuration mode and returns to privileged EXEC mode.

Configuring IS-IS IPv6 Client Support for BFD on All Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis**
4. **metric-style wide**
5. **address-family ipv6**
6. **multi-topology**
7. **bfd all-interfaces**
8. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router isis**

Example:

```
Device(config)# router isis
```

Enables the IS-IS routing protocol and enters router configuration mode.

Step 4 **metric-style wide**

Example:

```
Device(config-router)# metric-style wide
```

(Optional) Configures a device that is running IS-IS so that it generates and accepts only new-style type, length, value objects (TLVs).

Step 5 **address-family ipv6**

Example:

```
Device(config-router)# address-family ipv6
```

Enters address family configuration mode for configuring IS-IS routing sessions that use standard IPv6 address prefixes.

Step 6 **multi-topology**

Example:

```
Device(config-router-af)# multi-topology
```

(Optional) Enables multi-topology IS-IS for IPv6.

Step 7 **bfd all-interfaces**

Example:

```
Device(config-router-af)# bfd all-interfaces
```

Enables BFD for all interfaces participating in the routing process.

Step 8 **end**

Example:

```
Device(config-router-af)# end
```

Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for ISIS IPv6 Client for BFD

Example: IS-IS IPv6 Client Support for BFD on a Single Interface

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 6/0/0
Device(config-if)# ipv6 address 19:111:112::2/64
Device(config-if)# isis ipv6 bfd
Device(config-if)# end
```

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 6/0
Device(config-if)# ipv6 address 19:111:112::1/64
Device(config-if)# isis ipv6 bfd
Device(config-if)# end
```

Example: IS-IS IPv6 Client Support for BFD on All Interfaces

```
Device> enable
Device# configure terminal
Device(config)# router isis
Device(config-router)# metric-style wide
Device(config-router)# address-family ipv6
Device(config-router-af)# multi-topology
Device(config-router-af)# bfd all-interfaces
Device(config-router-af)# end
```

The following is a sample configuration where interface 0/0/7 of Router A is connected to interface 0/4/6 of router B.

Configuration for Router A

```
bfd-template single-hop BFDM
 interval min-tx 50 min-rx 50 multiplier 3
!
interface TenGigabitEthernet0/0/7
 ipv6 address 19:1:1::1/64
 ipv6 router isis
 bfd template BFDM
 isis ipv6 bfd
!
router isis
 net 49.0001.1720.1600.1001.00
!
```

Configuration on Router B

```
Router B

bfd-template single-hop BFDM
 interval min-tx 50 min-rx 50 multiplier 3
!
interface TenGigabitEthernet0/4/6
 ipv6 address 19:1:1::2/64
 ipv6 router isis
 bfd template BFDM
 isis ipv6 bfd
!
router isis
 net 49.0000.0000.0002.00
!
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
BFD commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
Configuring and monitoring IS-IS	"Configuring Integrated IS-IS" module of the *IP Routing Protocols Configuration Guide*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
IPv6 commands	*Cisco IOS IPv6 Command Reference*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

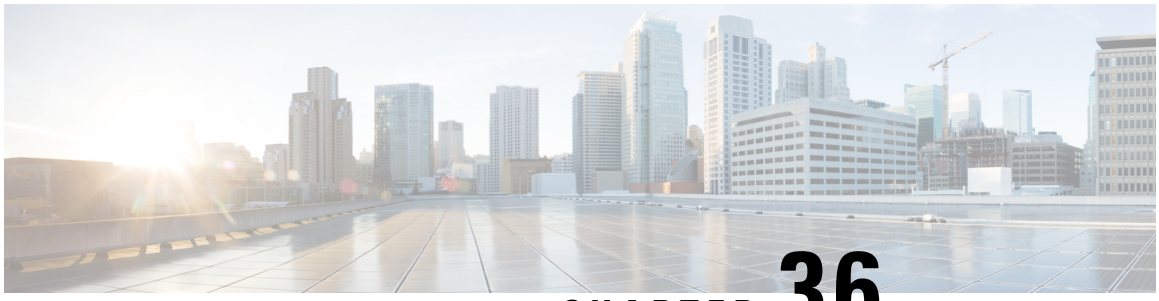
Feature Information for IS-IS IPv6 Client for BFD

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 43: Feature Information for IS-IS IPv6 Client for BFD

Feature Name	Releases	Feature Information
IS-IS IPv6 Client for BFD	15.1(1)SY 15.2(4)S 15.3(1)T	When BFD support is configured with IS-IS as a registered protocol with BFD, IS-IS receives forwarding path detection failure messages from BFD. The following commands were introduced or modified: **bfd all-interfaces**, **isis ipv6 bfd**.



CHAPTER 36

IS-IS Client for BFD C-Bit Support

The Bidirectional Forwarding Detection (BFD) protocol provides short-duration detection of failures in the path between adjacent forwarding engines while maintaining low networking overheads. The BFD IS-IS Client Support feature enables Intermediate System-to-Intermediate System (IS-IS) to use Bidirectional Forwarding Detection (BFD) support, which improves IS-IS convergence as BFD detection and failure times are faster than IS-IS convergence times in most network topologies. The IS-IS Client for BFD C-Bit Support feature enables the network to identify whether a BFD session failure is genuine or is the result of a control plane failure due to a router restart. When planning a router restart, you should configure this feature on all neighboring routers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for IS-IS Client for BFD C-Bit Support

- IS-IS must be running on all participating devices.
- The baseline parameters for BFD sessions must be configured on the interfaces that run BFD sessions to BFD neighbors.

Information About IS-IS Client for BFD C-Bit Support

IS-IS Restarts and BFD Sessions

The IS-IS Client for BFD C-Bit Support feature provides BFD with a way to signal to its peers whether the BFD implementation shares the same status as the control plane. When a neighboring router's control plane restarts, a BFD session failure may occur, which does not actually represent a true forwarding failure. If this happens, you do not want the neighbors of the restarting router to react to the BFD session failure.

IS-IS does not have protocol extensions that allow it to signal in advance that it will be restarting. This means that the system cannot distinguish between a real forwarding failure and a restart. The IS-IS Client for BFD C-Bit Support feature allows you to configure the device to ignore control-plane related BFD session failures. We recommend that you configure this feature on the neighbors of a restarting device just prior to the planned restart of that device and that you remove the configuration after the restart has been completed.

The table below shows how the control plane independent failure status received from BFD on a session down event impacts IS-IS handling of that event.

Table 44: Control Plane Failure and Session Down Events

IS-IS Check Control Plane Failure	BFD Control Plane Independent Failure Status	IS-IS Action on BFD session 'DOWN' Event
Enabled	True	Accept session DOWN
Enabled	False	Ignore session DOWN
Disabled	True	Accept session DOWN
Disabled	False	Accept session DOWN

How to Configure IS-IS Client for BFD C-Bit Support

Configuring IS-IS Client for BFD C-Bit Support

Perfrom this task to enable control plane failure checking.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis**
4. **bfd check-control-plane-failure**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** **Example:** `Device(config)# router isis`	Enables the IS-IS routing protocol and enters router configuration mode.
Step 4	**bfd check-control-plane-failure** **Example:** `Device(config-router)# bfd check-control-plane-failure`	Enables BFD control plane failure checking for the IS-IS routing protocol.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuration Examples for IS-IS Client for BFD C-Bit Support

Example: Configuring IS-IS Client for BFD C-Bit Support

The following example configures control plane failure detection on a router running the IS-IS protocol.

```
Device> enable
Device# configure terminal
Device(config)# router isis
Device(config-router)# bfd check-ctrl-plane-failure
Device(config-router)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
BFD commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: Protocol-Independent Command Reference*
Configuring and monitoring IS-IS	"Configuring Integrated IS-IS" module of the *Cisco IOS IP Routing Protocols Configuration Guide*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFC 5882	*Generic Application of Bidirectional Forwarding Detection (BFD)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

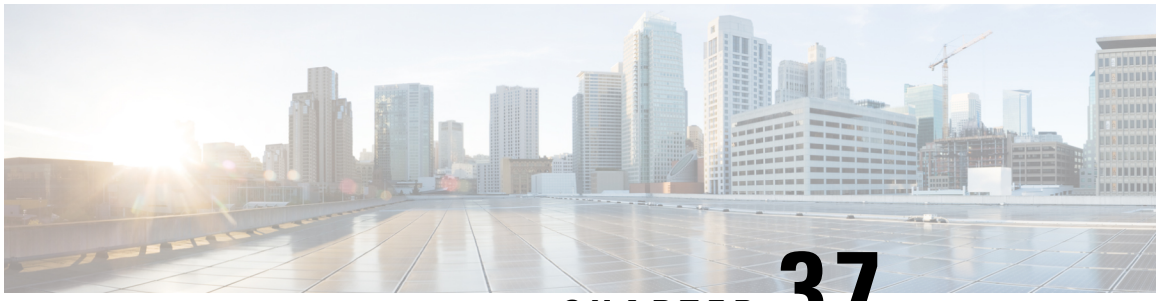
Feature Information for IS-IS Client for BFD C-Bit Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 45: Feature Information for IS-IS Client for BFD C-Bit Support

Feature Name	Releases	Feature Information
IS-IS Client for BFD C-Bit Support	15.1(1)SY 15.3(1)T	The IS-IS Client for BFD C-Bit Support feature enables the network to identify whether a BFD session failure is genuine or is the result of a control plane failure due to a router restart. The following command was introduced: **bfd check-ctrl-plane-failure**.



CHAPTER 37

BFD Dampening

The BFD Dampening feature introduces a configurable exponential delay mechanism to suppress the excessive effect of remote node reachability events flapping with Bidirectional Forwarding Detection (BFD). The BFD Dampening feature allows the network operator to automatically dampen a given BFD session to prevent excessive notification to the BFD clients, thus preventing unnecessary instability in the network. Configuring the BFD Dampening feature on a high-speed interface with routing clients improves the convergence time and stability throughout the network.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About BFD Dampening

Overview of BFD Dampening

Bidirectional Forwarding Detection (BFD) is a mechanism used by the routing protocols to quickly realize the reachability failures to their neighbors. When BFD detects a reachability status change of a neighbor, clients are notified immediately. Sometimes it might be critical to minimize changes in routing tables so as not to impact convergence, in case of any micro failure. An unstable link that flaps excessively can cause other devices in the network to consume substantial system processing resources, and it can cause routing protocols to lose synchronization with the state of the flapping link.

The BFD Dampening feature introduces a configurable exponential delay mechanism to suppress the excessive effect of remote node reachability events flapping with BFD. The BFD Dampening feature allows the network operator to automatically dampen a given BFD session to prevent excessive notification to the BFD clients, thus preventing unnecessary instability in the network. Dampening the notification to a BFD client suppresses BFD notification until the session under monitoring stops flapping and becomes stable.

Configuring the BFD Dampening feature, especially on a high-speed interface with routing clients, improves the convergence time and stability throughout the network. BFD dampening can be applied to all types of BFD sessions, including IPv4/single-hop/multihop, Multiprotocol Label Switching-Transport Profile (MPLS-TP), and Pseudo Wire (PW) Virtual Circuit Connection Verification (VCCV).

You can configure the BFD Dampening feature at the BFD template level (both single-hop and multihop templates). Dampening is applied to all the sessions that use the BFD template. If you do not want a session to be dampened, you should use a new BFD template without dampening for the new session. By default, the dampening functionality is not enabled on a template.

How to Configure BFD Dampening

Configuring BFD Dampening

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bfd-template multi-hop** *template-name*
4. **interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value*
5. **dampening** [*half-life-period reuse-threshold suppress-threshold max-suppress-time*]
6. **end**
7. **show bfd neighbors details**
8. **show bfd neighbors dampening**
9. **show bfd neighbors dampened**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bfd-template multi-hop** *template-name* **Example:**	Creates a Bidirectional Forwarding Detection (BFD) template and enters BFD configuration mode.

	Command or Action	Purpose
	`Device(config)# bfd-template multi-hop doctemplate`	
Step 4	**interval min-tx** *milliseconds* **min-rx** *milliseconds* **multiplier** *multiplier-value* **Example:** `Device(config-bfd)# interval min-tx 120 min-rx 100 multiplier 3`	Configures the transmit and receive intervals between BFD packets, and specifies the number of consecutive BFD control packets that must be missed before BFD declares that a peer is unavailable.
Step 5	**dampening** [*half-life-period reuse-threshold suppress-threshold max-suppress-time*] **Example:** `Device(config-bfd)# dampening 2 1000 3000 8`	Configures a device to dampen a flapping session.
Step 6	**end** **Example:** `Device(config-bfd)# end`	Exits BFD configuration mode and returns to privileged EXEC mode.
Step 7	**show bfd neighbors details** **Example:** `Device# show bfd neighbors details`	(Optional) Displays the listing of existing BFD adjacencies and the dampening information about the BFD sessions if BFD dampening is enabled for the session.
Step 8	**show bfd neighbors dampening** **Example:** `Device# show bfd neighbors dampening`	(Optional) Displays the dampening information about the BFD sessions configured with BFD dampening.
Step 9	**show bfd neighbors dampened** **Example:** `Device# show bfd neighbors dampened`	(Optional) Displays the dampening information about the BFD sessions that are currently dampened.

Configuration Examples for BFD Dampening

Example: Configuring BFD Dampening

The following example shows how to configure BFD dampening.

```
bfd-template multi-hop doctemplate
 interval min-tx 120 min-rx 100 multiplier 3
 dampening 2 1000 3000 8
```

Additional References for BFD Dampening

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BFD commands	Cisco IOS IP Routing: Protocol-Independent Command Reference
Bidirectional Forwarding Detection	*IP Routing BFD Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

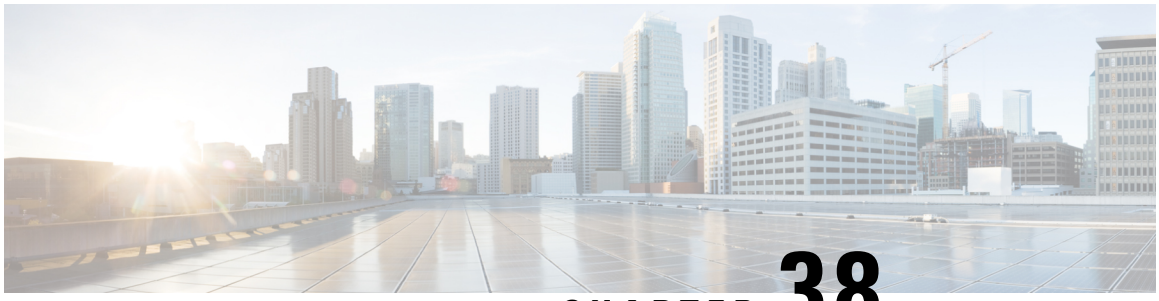
Feature Information for BFD Dampening

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 46: Feature Information for BFD Dampening

Feature Name	Releases	Feature Information
BFD Dampening	Cisco IOS XE Release 3.8S	The BFD Dampening feature introduces a configurable exponential delay mechanism to suppress the excessive effect of remote node reachability events flapping with BFD. This feature also allows the network operator to automatically dampen a given BFD session to prevent excessive notification to the BFD clients, thus preventing unnecessary instability in the network. The following commands were introduced or modified: **dampening (bfd)** and **show bfd neighbors**.



CHAPTER 38

Bidirectional Forwarding Detection on Link Aggregation Group Bundle

The Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Bundle feature enables users to configure individual BFD sessions on each LAG member interface.

Feature Information for Bidirectional Forwarding Detection on Link Aggregation Group Bundle

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 47: Feature Information for Bidirectional Forwarding Detection on Link Aggregation Group Bundle

Feature Name	Releases	Feature Information
Micro BFD Support with LACP	Cisco IOS XE Bengaluru 17.4.1a	Micro-BFD, which is supported for the physical member-links within a port-channel is now configured to receive BFD events and to create BFD sessions per member-link. The member-links are able to receive BFD events after you enable Micro-BFD for the port-channel member-links.

Feature Name	Releases	Feature Information
Bidirectional Forwarding Detection on Link Aggregation Group Bundle	Cisco IOS XE Fuji 16.8.1	The Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Bundle feature enables users to configure individual BFD sessions on each LAG member interface.

Information About Bidirectional Forwarding Detection on Link Aggregation Group Bundle

The Bidirectional Forwarding Detection (BFD) enhancement to address per-link efficiency feature enables users to configure individual BFD sessions on each Link Aggregation Group (LAG) member interface. With this enhancement BFD sessions run on each member link of the port-channel. BFD sessions running on member links of the port-channel are called as micro BFD sessions. Micro BFD sessions are supported for both LACP and non-LACP based-port channels. If BFD detects a fault in the bidirectional path between two forwarding engines that includes interfaces and data links, the member link is removed from the forwarding table. This mechanism delivers faster failure detection as the BFD sessions are created on individual port-channel interface. Users can configure BFD over main port-channel interface, that will monitor the bandwidth consumption of LAG by using a micro BFD session for each member. If any member port goes down the port is removed from the forwarding table and this prevents a null route for that member. Micro BFD works only when it is configured on all the members of port-channel. The logical BFD session takes less aggressive timers than the BFD on LAG sessions, whether it is configured on port-channel or port-channel sub-interfaces.

LAG combines multiple physical links into a single logical link that helps in providing higher bandwidth and better resiliency. If the physical member links fails, the aggregate logical link can continue to forward traffic over the remaining operational physical member links.

With the support of micro BFD feature, port channel manager considers the state of micro BFD sessions to determine the state of the port channel interface. Port channel implementation provides minimum links (lacp min-links) configuration to ensure bandwidth availability by making a port channel usable or unusable based on whether configured number of ports are available or not. The detection of micro BFD happens only when Link Aggregation Control Protocol (LACP) in COLLECTIING_DISTRIBUTION is in active state. Maximum member port supported for LACP mode per port-channel varies from one platform to another. For example, ASR 1000 supports 14 port-channels.

The goal of micro BFD sessions are:

- Run BFD session over each LAG member link.
- Verify link continuity for each member link.
- Allow BFD to control the LAG member link to be part of the L2 load-balancing table of the LAG interface in the presence or absence of LACP.

Note When a member-link receives a BFD_DOWN event, it is removed from the port-channel and the member link is added back to the port-channel only when a BFD_UP event is received on that member-link.

Restrictions for Bidirectional Forwarding Detection on Link Aggregation Group Bundle

- Micro BFD sessions are not supported on port-channel sub interfaces.
- Echo functionality is not supported on micro BFD sessions.
- BFD supports only IPv4.
- Micro BFD hardware offload is not supported.
- Micro BFD sessions are not supported on partial member links of bundled port-channel.

How to Configure Bidirectional Forwarding Detection on Link Aggregation Group Bundle

Before you configure BFD template, ensure that BFD is enabled.

Configuring BFD Template

```
Device(config)# bfd-template single-hop testing
Device(config-bfd)# interval min-tx 50 min-rx 50 multiplier 3
Device(config-bfd)# end
```

The time interval specified can be up to 9999 milliseconds.

Applying Template to Port-Channel Interface

```
Device(config)#interface port-channel 60
Device(config-if)#port-channel bfd destination ipv4 192.0.2.1 testing
Device(config-if)#ip address 192.0.2.2 255.255.255.0
Device(config-if)#no shutdown
Device(config-if)#end
```

Note Ensure that you run no shut on port-channel command, if you change the IP address of port-channel provided the member links are already added to port channel and Micro BFD is configured on port-channel.

Adding Member Ports to Port-Channel Group

Perform the following steps to add member ports to port-channel group:

```
Device(config)#interface Gi0/0/0
Device(config-if)#channel-group 60 mode active
Device(config-if)#no shutdown
Device(config-if)#end
```

Verifying Bidirectional Forwarding Detection on Link Aggregation Group Bundle

Verifying Port Bundle State for BFD

```
show etherchannel summary

Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

 Number of channel-groups in use: 1

Number of aggregators:           1

Group  Port-channel  Protocol    Ports

------+-------------+-----------+-----------------------------------------------

60 Po60(RU) LACP  Gi0/0/0(bndl)

 RU - L3 port-channel UP State

SU - L2 port-channel UP state

P/bndl -  Bundled

S/susp  - Suspended

R1#
```

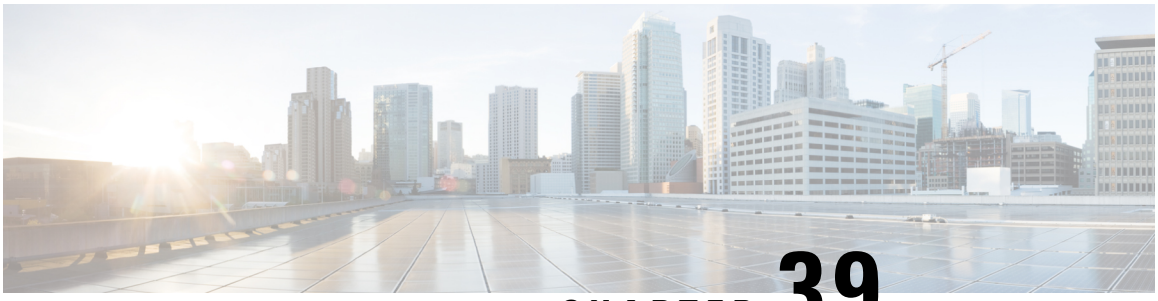
Verifying Micro BFD Sessions

```
Device#show bfd neighbors interface Gi0/0/0 details

Port Channel IPv4 Sessions

NeighAddr                              LD/RD         RH/RS     State     Int    Parent Int

192.0.2.2                             4121/4120      Up        Up        Gi0/0/0     Po60

Session state is UP and not using echo function.

Session Host: Software

OurAddr: 192.0.2.1

Handle: 1

Local Diag: 0, Demand mode: 0, Poll bit: 0
```

```
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3

Received MinRxInt: 50000, Received Multiplier: 3

Holddown (hits): 145(0), Hello (hits): 50(46)

Rx Count: 46, Rx Interval (ms) min/max/avg: 1/50/43 last: 5 ms ago

Tx Count: 47, Tx Interval (ms) min/max/avg: 1/50/41 last: 26 ms ago

Elapsed time watermarks: 0 0 (last: 0)

Registered protocols: PochIPv4

Template: testing

Uptime: 00:00:01

Last packet: Version: 1                  - Diagnostic: 0

             State bit: Up               - Demand bit: 0

             Poll bit: 0                 - Final bit: 0

             C bit: 0

             Multiplier: 3               - Length: 24

             My Discr.: 4120             - Your Discr.: 4121

             Min tx interval: 50000      - Min rx interval: 50000

             Min Echo interval: 0
Device(config)#do show etherchannel summary
Flags:  D - down        P/bndl - bundled in port-channel
        I - stand-alone s/susp - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 3
Number of aggregators:           3

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1 Po1(RU)  LACP  Gi2/1/0(bndl) Gi2/1/1(bndl) Gi2/1/2(bndl)
10 Po10(RU)    Gi0/1/2(P)
20 Po20(RU)  LACP  Gi0/1/3(bndl) Gi0/1/4(bndl)

RU - L3 port-channel UP State
SU - L2 port-channel UP state
P/bndl -  Bundled
S/susp  - Suspended
```

PART III

BGP



CHAPTER 39

Cisco BGP Overview

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco software implementation of BGP version 4 includes support for 4-byte autonomous system numbers and multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks Version 4 (VPNv4), Connectionless Network Services (CLNS), and Layer 2 VPN (L2VPN). This module contains conceptual material to help you understand how BGP is implemented in Cisco software.

Prerequisites for Cisco BGP

This document assumes knowledge of CLNS, IPv4, IPv6, multicast, VPNv4, and Interior Gateway Protocols (IGPs). The amount of knowledge required for each technology is dependent on your deployment.

Restrictions for Cisco BGP

A router that runs Cisco software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple concurrent BGP address family and subaddress family configurations.

Information About Cisco BGP

BGP Version 4

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco software implementation of BGP version 4 includes multiprotocol extensions to allow BGP to carry routing

information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks version 4 (VPNv4), and Connectionless Network Services (CLNS).

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. For more details about connecting to external BGP peers, see the "Connecting to a Service Provider Using External BGP" chapter.

Although BGP is referred to as an exterior gateway protocol (EGP), many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions. For more details about internal BGP peers, see the "Configuring Internal BGP Features" chapter of the *Cisco IOS IP Routing Configuration Guide*.

Note BGP requires more configuration than other routing protocols and the effects of any configuration changes must be fully understood. Incorrect configuration can create routing loops and negatively impact normal network operation.

BGP Version 4 Functional Overview

BGP is an interdomain routing protocol designed to provide loop-free routing links between organizations. BGP is designed to run over a reliable transport protocol; it uses TCP (port 179) as the transport protocol because TCP is a connection-oriented protocol. The destination TCP port is assigned 179, and the local port is assigned a random port number. Cisco software supports BGP version 4 and it is this version that has been used by Internet service providers (ISPs) to help build the Internet. RFC 1771 introduced and discussed a number of new BGP features to allow the protocol to scale for Internet use. RFC 2858 introduced multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families, including IPv4, IPv6, and CLNS.

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. Although BGP is referred to as an exterior gateway protocol (EGP), many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

BGP uses a path-vector routing algorithm to exchange network reachability information with other BGP-speaking networking devices. Network reachability information is exchanged between BGP peers in routing updates. Network reachability information contains the network number, path-specific attributes, and the list of autonomous system numbers that a route must transit to reach a destination network. This list is contained in the AS-path attribute. BGP prevents routing loops by rejecting any routing update that contains the local autonomous system number because this indicates that the route has already traveled through that autonomous system and a loop would therefore be created. The BGP path-vector routing algorithm is a combination of the distance-vector routing algorithm and the AS-path loop detection.

BGP selects a single path, by default, as the best path to a destination host or network. The best path selection algorithm analyzes path attributes to determine which route is installed as the best path in the BGP routing table. Each path carries well-known mandatory, well-known discretionary, and optional transitive attributes that are used in BGP best path analysis. Cisco software provides the ability to influence BGP path selection by altering some of these attributes using the command-line interface (CLI.) BGP path selection can also be

influenced through standard BGP policy configuration. For more details about using BGP to influence path selection and configuring BGP policies to filter traffic, see the "BGP 4 Prefix Filter and Inbound Route Maps" module and the "BGP Prefix-Based Outbound Route Filtering" module.

BGP uses the best-path selection algorithm to find a set of equally good routes. These routes are the potential multipaths. In Cisco IOS Release 12.2(33)SRD and later releases, when there are more equally good multipaths available than the maximum permitted number, the oldest paths are selected as multipaths.

BGP can be used to help manage complex internal networks by interfacing with Interior Gateway Protocols (IGPs). Internal BGP can help with issues such as scaling the existing IGPs to match the traffic demands while maintaining network efficiency.

Note BGP requires more configuration than other routing protocols and the effects of any configuration changes must be fully understood. Incorrect configuration can create routing loops and negatively impact normal network operation.

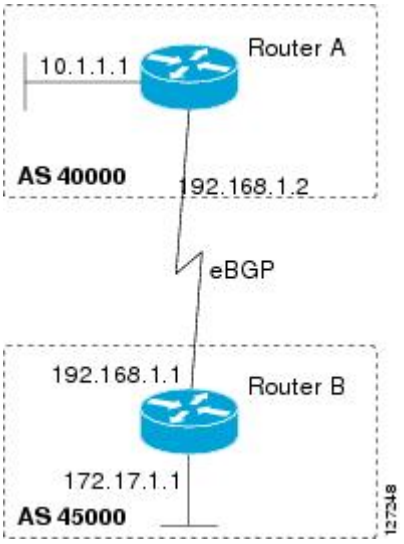
BGP Autonomous Systems

An autonomous system is a network controlled by a single technical administration entity. BGP autonomous systems are used to divide global external networks into individual routing domains where local routing policies are applied. This organization simplifies routing domain administration and simplifies consistent policy configuration. Consistent policy configuration is important to allow BGP to efficiently process routes to destination networks.

Each routing domain can support multiple routing protocols. However, each routing protocol is administered separately. Other routing protocols can dynamically exchange routing information with BGP through redistribution. Separate BGP autonomous systems dynamically exchange routing information through eBGP peering sessions. BGP peers within the same autonomous system exchange routing information through iBGP peering sessions.

The figure below illustrates two routers in separate autonomous systems that can be connected using BGP. Router A and Router B are ISP routers in separate routing domains that use public autonomous system numbers. These routers carry traffic across the Internet. Router A and Router B are connected through eBGP peering sessions.

Figure 32: BGP Topology with Two Autonomous Systems



Each public autonomous system that directly connects to the Internet is assigned a unique number that identifies both the BGP routing process and the autonomous system.

BGP Autonomous System Number Formats

Prior to January 2009, BGP autonomous system numbers that were allocated to companies were two-octet numbers in the range from 1 to 65535 as described in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*. Due to increased demand for autonomous system numbers, the Internet Assigned Number Authority (IANA) will start in January 2009 to allocate four-octet autonomous system numbers in the range from 65536 to 4294967295. RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*, documents three methods of representing autonomous system numbers. Cisco has implemented the following two methods:

- Asplain--Decimal value notation where both 2-byte and 4-byte autonomous system numbers are represented by their decimal value. For example, 65526 is a 2-byte autonomous system number and 234567 is a 4-byte autonomous system number.
- Asdot--Autonomous system dot notation where 2-byte autonomous system numbers are represented by their decimal value and 4-byte autonomous system numbers are represented by a dot notation. For example, 65526 is a 2-byte autonomous system number and 1.169031 is a 4-byte autonomous system number (this is dot notation for the 234567 decimal number).

For details about the third method of representing autonomous system numbers, see RFC 5396.

Asdot Only Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases, the 4-octet (4-byte) autonomous system numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte autonomous system numbers the asdot format includes a period which is a special character in regular expressions. A backslash must be entered before the period for example, 1\.14, to ensure the regular expression match does not fail. The table below shows the format in which 2-byte and 4-byte autonomous system numbers are configured, matched in regular expressions, and displayed in **show** command output in Cisco IOS images where only asdot formatting is available.

Table 48: Asdot Only 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte autonomous system numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte autonomous system numbers are written in the asplain format. If you want to change the default **show** command output to display 4-byte autonomous system numbers in the asdot format, use the **bgp asnotation dot** command under router configuration mode. When the asdot format is enabled as the default, any regular expressions to match 4-byte autonomous system numbers must be written using the asdot format, or the regular expression match will fail. The tables below show that although you can configure 4-byte autonomous system numbers in either asplain or asdot format, only one format is used to display **show** command output and control 4-byte autonomous system number matching for regular expressions, and the default is asplain format. To display 4-byte autonomous system numbers in **show** command output and to control matching for regular expressions in the asdot format, you must configure the **bgp asnotation dot** command. After enabling the **bgp asnotation dot** command, a hard reset must be initiated for all BGP sessions by entering the **clear ip bgp *** command.

Note If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The **show** command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

Table 49: Default Asplain 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 50: Asdot 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private Autonomous System Numbers

In Cisco IOS Release 12.0(32)S12, 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 12.4(24)T, and later releases, the Cisco implementation of BGP supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use*, describes new reserved autonomous system numbers for documentation purposes. Use of the reserved numbers allow configuration examples to be accurately documented and avoids conflict with production networks if these configurations are literally copied. The reserved numbers are documented in the IANA autonomous system number registry. Reserved 2-byte autonomous system numbers are in the contiguous block, 64496 to 64511 and reserved 4-byte autonomous system numbers are from 65536 to 65551 inclusive.

Private 2-byte autonomous system numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private autonomous system numbers can be used for internal routing domains but must be translated for traffic that is routed out to the Internet. BGP should not be configured to advertise private autonomous system numbers to external networks. Cisco IOS software does not remove private autonomous system numbers from routing updates by default. We recommend that ISPs filter private autonomous system numbers.

Note Autonomous system number assignment for public and private networks is governed by the IANA. For information about autonomous-system numbers, including reserved number assignment, or to apply to register an autonomous system number, see the following URL: http://www.iana.org/.

Classless Interdomain Routing

BGP version 4 supports classless interdomain routing (CIDR). CIDR eliminates classful network boundaries, providing more efficient usage of the IPv4 address space. CIDR provides a method to reduce the size of routing tables by configuring aggregate routes (or supernets). CIDR processes a prefix as an IP address and bit mask (bits are processed from left to right) to define each network. A prefix can represent a network, subnetwork, supernet, or single host route.

For example, using classful IP addressing, the IP address 192.168.2.1 is defined as a single host in the Class C network 192.168.2.0. Using CIDR, the IP address can be shown as 192.168.2.1/16, which defines a network (or supernet) of 192.168.0.0.

CIDR is enabled by default for all routing protocols in Cisco software. Enabling CIDR affects how packets are forwarded, but it does not change the operation of BGP.

Multiprotocol BGP

Cisco software supports multiprotocol BGP extensions as defined in RFC 2858, *Multiprotocol Extensions for BGP-4*. The extensions introduced in this RFC allow BGP to carry routing information for multiple network-layer protocols, including CLNS, IPv4, IPv6, and VPNv4. These extensions are backward-compatible to enable routers that do not support multiprotocol extensions to communicate with those routers that do support multiprotocol extensions. Multiprotocol BGP carries routing information for multiple network-layer protocols and IP multicast routes. BGP carries different sets of routes depending on the protocol. For example,

BGP can carry one set of routes for IPv4 unicast routing, one set of routes for IPv4 multicast routing, and one set of routes for MPLS VPNv4 routes.

Note A multiprotocol BGP network is backward-compatible with a BGP network, but BGP peers that do not support multiprotocol extensions cannot forward routing information, such as address family identifier information, that the multiprotocol extensions carry.

Benefits of Using Multiprotocol BGP Versus BGP

In complex networks with multiple network layer protocols, multiprotocol BGP must be used. In less complex networks we recommend using multiprotocol BGP because it offers the following benefits:

- All of the BGP commands and routing policy capabilities of BGP can be applied to multiprotocol BGP.
- A network can carry routing information for multiple network layer protocol address families (for example, IP Version 4 or VPN Version 4) as specified in RFC 1700, *Assigned Numbers*.
- A network can support incongruent unicast and multicast topologies.
- A multiprotocol BGP network is backward compatible because the routers that support the multiprotocol extensions can interoperate with routers that do not support the extensions.

In summary, multiprotocol BGP support for multiple network layer protocol address families provides a flexible and scalable infrastructure that allows you to define independent policy and peering configurations on a per-address family basis.

Multiprotocol BGP Extensions for IP Multicast

The routes associated with multicast routing are used by the Protocol Independent Multicast (PIM) feature to build data distribution trees. Multiprotocol BGP is useful when you want a link that is dedicated to multicast traffic, perhaps to limit which resources are used for which traffic. For example, you want all multicast traffic exchanged at one network access point (NAP). Multiprotocol BGP allows you to have a unicast routing topology different from a multicast routing topology, which allows you more control over your network and resources.

In BGP, the only way to perform interdomain multicast routing is to use the BGP infrastructure that is in place for unicast routing. If the routers are not multicast-capable, or if there are differing policies about where multicast traffic should flow, multicast routing cannot be supported without multiprotocol BGP.

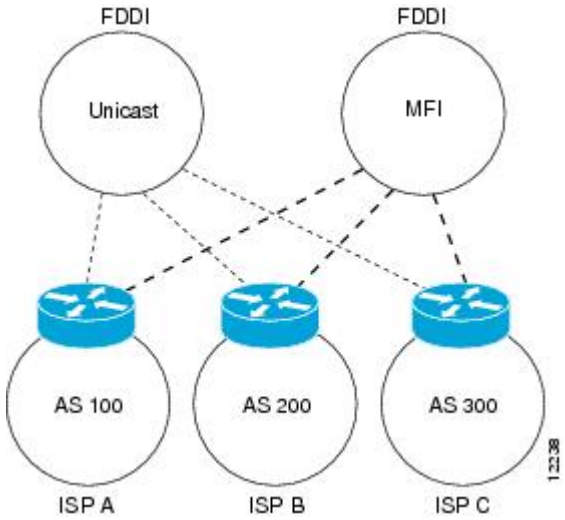
A multicast routing protocol, such as PIM, uses both the multicast and unicast BGP database to source the route, perform Reverse Path Forwarding (RPF) lookups for multicast-capable sources, and build a multicast distribution tree (MDT). The multicast table is the primary source for the router, but if the route is not found in the multicast table, the unicast table is searched. Although multicast can be performed with unicast BGP, multicast BGP routes allow an alternative topology to be used for RPF.

It is possible to configure BGP peers that exchange both unicast and multicast Network Layer Reachability Information (NLRI) where multiprotocol BGP routes can be redistributed into BGP. Multiprotocol extensions, however, will be ignored by any peers that do not support multiprotocol BGP. When PIM builds a multicast distribution tree through a unicast BGP network (because the route through the unicast network is the most attractive), the RPF check may fail, preventing the MDT from being built. If the unicast network runs multiprotocol BGP, peering can be configured using the appropriate multicast address family. The multicast

address family configuration enables multiprotocol BGP to carry the multicast information and the RPF lookup will succeed.

The figure below illustrates a simple example of unicast and multicast topologies that are incongruent; these topologies cannot exchange information without implementing multiprotocol BGP. Autonomous systems 100, 200, and 300 are each connected to two NAPs that are FDDI rings. One is used for unicast peering (and therefore the exchanging of unicast traffic). The Multicast Friendly Interconnect (MFI) ring is used for multicast peering (and therefore the exchanging of multicast traffic). Each router is unicast- and multicast-capable.

Figure 33: Incongruent Unicast and Multicast Routes

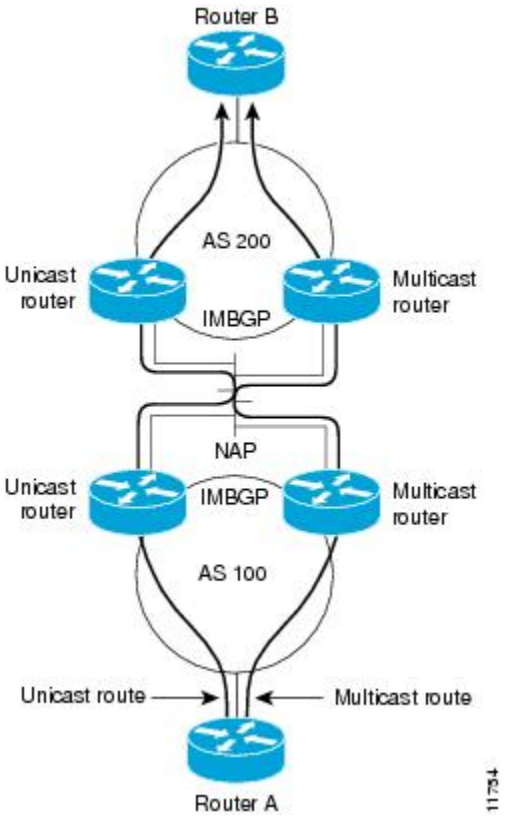


The figure below is a topology of unicast-only routers and multicast-only routers. The two routers on the left are unicast-only routers (that is, they do not support or are not configured to perform multicast routing). The two routers on the right are multicast-only routers. Routers A and B support both unicast and multicast routing. The unicast-only and multicast-only routers are connected to a single NAP.

In the figure below, only unicast traffic can travel from Router A to the unicast routers to Router B and back. Multicast traffic could not flow on that path, because multicast routing is not configured on the unicast routers and therefore the BGP routing table does not contain any multicast routes. On the multicast routers, multicast routes are enabled and BGP builds a separate routing table to hold the multicast routes. Multicast traffic uses the path from Router A to the multicast routers to Router B and back.

The figure below illustrates a multiprotocol BGP environment with a separate unicast route and multicast route from Router A to Router B. Multiprotocol BGP allows these routes to be noncongruent. Both of the autonomous systems must be configured for internal multiprotocol BGP (labeled “IMBGP” in the figure).

Figure 34: Multicast BGP Environment



For more information about IP multicast, see the “Configuring IP Multicast” configuration library.

NLRI Configuration CLI

BGP was designed to carry only unicast IPv4 routing information. BGP configuration used the Network NLRI format CLI in Cisco software. The NLRI format offers only limited support for multicast routing information and does not support multiple network layer protocols. We do not recommend using NLRI format CLI for BGP configuration.

Using the BGP hybrid CLI feature, you can configure commands in the address family VPNv4 format and save these command configurations without modifying an existing NLRI formatted configuration. If you want to use other address family configurations such as IPv4 unicast or multicast, then you must upgrade the configuration using the **bgp upgrade-cli** command.

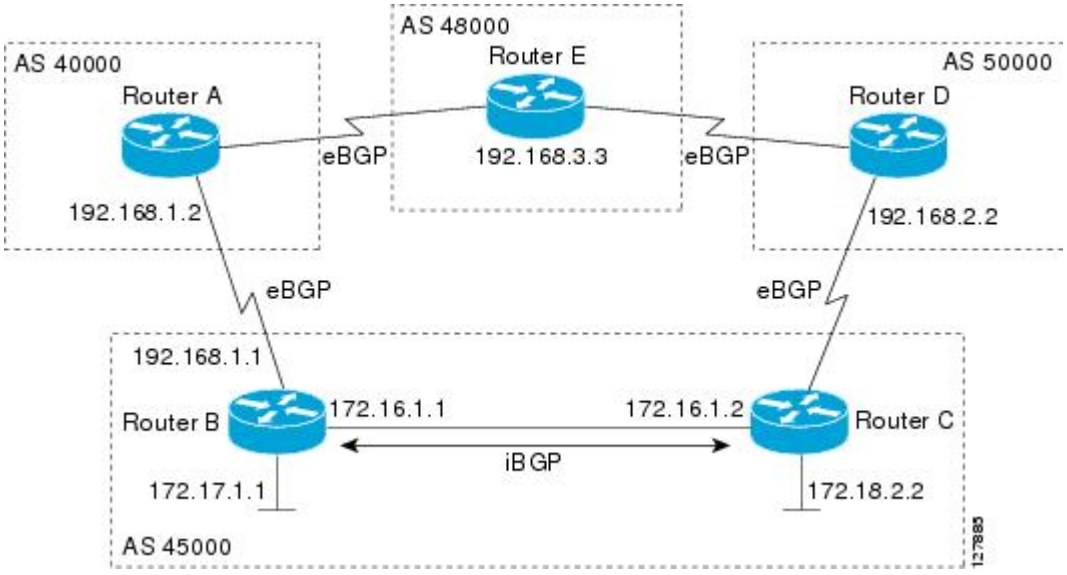
For more details about using BGP hybrid CLI commands, see the "Configuring a Basic BGP Network” module. See the “Multiprotocol BGP” and “Cisco BGP Address Family Model” sections for more information about address family configuration format and the limitations of the NLRI CLI format.

Cisco BGP Address Family Model

The Cisco BGP address family identifier (AFI) model was introduced with multiprotocol BGP and is designed to be modular and scalable, and to support multiple AFI and subsequent address family identifier (SAFI) configurations. Networks are increasing in complexity and many companies are now using BGP to connect

to many autonomous systems, as shown in the network topology in the figure below. Each of the separate autonomous systems shown in the figure below may be running several routing protocols such as Multiprotocol Label Switching (MPLS) and IPv6 and require both unicast and multicast routes to be transported via BGP.

Figure 35: BGP Network Topology for Multiple Address Families



The Cisco BGP AFI model introduced new command-line interface (CLI) commands supported by a new internal structure. Multiprotocol BGP carries routing information for multiple network layer protocols and IP multicast routes. This routing information is carried in the AFI model as appended BGP attributes (multiprotocol extensions). Each address family maintains a separate BGP database, which allows you to configure BGP policy on per-address family basis. SAFI configurations are subsets of the parent AFI. SAFIs can be used to refine BGP policy configurations.

The AFI model was created because of scalability limitations of the NLRI format. A router that is configured in NLRI format has IPv4 unicast but limited multicast capabilities. Networks that are configured in the NLRI format have the following limitations:

- No support for AFI and SAFI configuration information. Many new BGP (and other protocols such as MPLS) features are supported only in AFI and SAFI configuration modes and cannot be configured in NLRI configuration modes.
- No support for IPv6. A router that is configured in the NLRI format cannot establish peering with an IPv6 neighbor.
- Limited support for multicast interdomain routing and incongruent multicast and unicast topologies. In the NLRI format, not all configuration options are available and there is no support for VPNv4. The NLRI format configurations can be more complex than configurations that support the AFI model. If the routers in the infrastructure do not have multicast capabilities, or if policies differ as to where multicast traffic is configured to flow, multicast routing cannot be supported.

The AFI model in multiprotocol BGP supports multiple AFIs and SAFIs, all NLRI-based commands and policy configurations, and is backward compatible with routers that support only the NLRI format. A router that is configured using the AFI model has the following features:

- AFI and SAFI information and configurations are supported. A router that is configured using the AFI model can carry routing information for multiple network layer protocol address families (for example, IPv4 and IPv6).
- AFI configuration is similar in all address families, making the CLI syntax easier to use than the NLRI format syntax.
- All BGP routing policy capabilities and commands are supported.
- Congruent unicast and multicast topologies that have different policies (BGP filtering configurations) are supported, as are incongruent multicast and unicast topologies.
- CLNS is supported.
- Interoperation between routers that support only the NLRI format (AFI-based networks are backward compatible) is supported. This includes both IPv4 unicast and multicast NLRI peers.
- Virtual Private Networks (VPNs) and VPN routing and forwarding (VRF) instances are supported. Unicast IPv4 for VRFs can be configured from a specific address family IPv4 VRF; this configuration update is integrated into the BGP VPNv4 database.

Within a specific address family configuration mode, the question mark (?) online help function can be used to display supported commands. The BGP commands supported in address family configuration mode configure the same functionality as the BGP commands supported in router configuration mode; however, the BGP commands in router configuration mode configure functionality only for the IPv4 unicast address prefix. To configure BGP commands and functionality for other address family prefixes (for example, the IPv4 multicast or IPv6 unicast address prefixes), you must enter address family configuration mode for those address prefixes.

The BGP address family model consists of four address families in Cisco IOS software; IPv4, IPv6, CLNS, and VPNv4. In Cisco IOS Release 12.2(33)SRB, and later releases, support for the L2VPN address family was introduced, and within the L2VPN address family the VPLS SAFI is supported. Within the IPv4 and IPv6 address families, SAFIs such as Multicast Distribution Tree (MDT), tunnel, and VRF exist. The table below shows the list of SAFIs supported by Cisco IOS software. To ensure compatibility between networks running all types of AFI and SAFI configuration, we recommend configuring BGP on Cisco IOS devices using the multiprotocol BGP address family model.

Table 51: SAFIs Supported by Cisco IOS Software

SAFI Field Value	Description	Reference
1	NLRI used for unicast forwarding.	RFC 2858
2	NLRI used for multicast forwarding.	RFC 2858
3	NLRI used for both unicast and multicast forwarding.	RFC 2858
4	NLRI with MPLS labels.	RFC 3107
64	Tunnel SAFI.	draft-nalawade-kapoor-tunnel- safi -01.txt
65	Virtual Private LAN Service (VPLS).	—
66	BGP MDT SAFI.	draft-nalawade-idr-mdt-safi-00.txt

SAFI Field Value	Description	Reference
128	MPLS-labeled VPN address.	RFC-ietf-l3vpn-rfc2547bis-03.txt

IPv4 Address Family

The IPv4 address family is used to identify routing sessions for protocols such as BGP that use standard IP version 4 address prefixes. Unicast or multicast address prefixes can be specified within the IPv4 address family. Routing information for address family IPv4 unicast is advertised by default when a BGP peer is configured unless the advertisement of unicast IPv4 information is explicitly turned off.

VRF instances can also be associated with IPv4 AFI configuration mode commands.

In Cisco IOS Release 12.0(28)S, the tunnel SAFI was introduced to support multipoint tunneling IPv4 routing sessions. The tunnel SAFI is used to advertise the tunnel endpoints and the SAFI specific attributes that contain the tunnel type and tunnel capabilities. Redistribution of tunnel endpoints into the BGP IPv4 tunnel SAFI table occurs automatically when the tunnel address family is configured. However, peers need to be activated under the tunnel address family before the sessions can exchange tunnel information.

In Cisco IOS Release 12.0(29)S, the multicast distribution tree (MDT) SAFI was introduced to support multicast VPN architectures. The MDT SAFI is a transitive multicast capable connector attribute that is defined as an IPv4 address family in BGP. The MDT address family session operates as a SAFI under the IPv4 multicast address family, and is configured on provider edge (PE) routers to establish VPN peering sessions with customer edge (CE) routers that support inter-AS multicast VPN peering sessions.

IPv6 Address Family

The IPv6 address family is used to identify routing sessions for protocols such as BGP that use standard IPv6 address prefixes. Unicast or multicast address prefixes can be specified within the IPv6 address family.

Note Routing information for address family IPv4 unicast is advertised by default when you configure a BGP peer unless you explicitly turn off the advertisement of unicast IPv4 information.

CLNS Address Family

The CLNS address family is used to identify routing sessions for protocols such as BGP that use standard network service access point (NSAP) address prefixes. Unicast address prefixes are the default when NSAP address prefixes are configured.

CLNS routes are used in networks where CLNS addresses are configured. This is typically a telecommunications Data Communications Network (DCN). Peering is established using IP addresses, but update messages contain CLNS routes.

For more details about configuring BGP support for CLNS, which provides the ability to scale CLNS networks, see the “Configuring Multiprotocol BGP (MP-BGP) support for CLNS” module.

VPNv4 Address Family

The VPNv4 multicast address family is used to identify routing sessions for protocols such as BGP that use standard VPN Version 4 address prefixes. Unicast address prefixes are the default when VPNv4 address prefixes are configured. VPNv4 routes are the same as IPv4 routes, but VPNv4 routes have a route descriptor (RD) prepended that allows replication of prefixes. It is possible to associate every different RD with a different VPN. Each VPN needs its own set of prefixes.

Companies use an IP VPN as the foundation for deploying or administering value-added services including applications and data hosting network commerce, and telephony services to business customers.

In private LANs, IP-based intranets have fundamentally changed the way companies conduct their business. Companies are moving their business applications to their intranets to extend over a WAN. Companies are also addressing the needs of their customers, suppliers, and partners by using extranets (an intranet that encompasses multiple businesses). With extranets, companies reduce business process costs by facilitating supply-chain automation, electronic data interchange (EDI), and other forms of network commerce. To take advantage of this business opportunity, service providers must have an IP VPN infrastructure that delivers private network services to businesses over a public infrastructure.

VPNs, when used with MPLS, allow several sites to transparently interconnect through a service provider's network. One service provider network can support several different IP VPNs. Each of these appears to its users as a private network, separate from all other networks. Within a VPN, each site can send IP packets to any other site in the same VPN. Each VPN is associated with one or more VPN VRFs. VPNv4 routes are a superset of routes from all VRFs, and route injection is done per VRF under the specific VRF address family. The router maintains a separate routing and Cisco Express Forwarding (CEF) table for each VRF. This prevents information from being sent outside the VPN and allows the same subnet to be used in several VPNs without causing duplicate IP address problems. The router using BGP distributes the VPN routing information using the BGP extended communities.

The VPN address space is isolated from the global address space by design. BGP distributes reachability information for VPN-IPv4 prefixes for each VPN using the VPNv4 multiprotocol extensions to ensure that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

RFC 3107 specifies how to add label information to multiprotocol BGP address families using a SAFI. The Cisco IOS implementation of MPLS uses RFC 3107 to provide support for sending IPv4 routes with a label. VPNv4 routes implicitly have a label associated with each route.

L2VPN Address Family

L2VPN is defined as a secure network that operates inside an unsecured network by using an encryption technology such as IP security (IPsec) or Generic Routing Encapsulation (GRE). The L2VPN address family is configured under BGP routing configuration mode, and within the L2VPN address family the VPLS subsequent address family identifier (SAFI) is supported.

BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN routing information base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. Prefix and path information is stored in the L2VPN database, allowing BGP to make best-path decisions. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the setting up of L2VPN services, which are an integral part of the Cisco IOS Virtual Private LAN Service (VPLS) feature. VPLS enables flexibility in deploying services

by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP MPLS network. For more details about VPLS, see the "VPLS Autodiscovery: BGP Based" feature.

Under L2VPN address family the following BGP command-line interface (CLI) commands are supported:

- **bgp scan-time**
- **bgp nexthop**
- **neighbor activate**
- **neighbor advertisement-interval**
- **neighbor allowas-in**
- **neighbor capability**
- **neighbor inherit**
- **neighbor peer-group**
- **neighbor maximum-prefix**
- **neighbor next-hop-self**
- **neighbor next-hop-unchanged**
- **neighbor remove-private-as**
- **neighbor route-map**
- **neighbor route-reflector-client**
- **neighbor send-community**
- **neighbor soft-reconfiguration**
- **neighbor soo**
- **neighbor weight**

Note For route reflectors using L2VPNs, the **neighbor next-hop-self** and **neighbor next-hop-unchanged** commands are not supported.

For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration. All other route map commands are supported.

BGP multipaths and confederations are not supported under the L2VPN address family.

For details on configuring BGP under the L2VPN address family, see the "BGP Support for the L2VPN Address Family" module.

BGP CLI Removal Considerations

BGP CLI configuration can become quite complex even in smaller BGP networks. If you need to remove any CLI configuration, you must consider all the implications of removing the CLI. Analyze the current running

configuration to determine the current BGP neighbor relationships, any address family considerations, and even other routing protocols that are configured. Many BGP CLI commands affect other parts of the CLI configuration. For example, in the following configuration, a route map is used to match a BGP autonomous system number and then set the matched routes with another autonomous system number for EIGRP:

```
route-map bgp-to-eigrp permit 10
 match tag 50000
 set tag 65000
```

BGP neighbors in three different autonomous systems are configured and activated:

```
router bgp 45000
 bgp log-neighbor-changes
 address-family ipv4
  neighbor 172.16.1.2 remote-as 45000
  neighbor 192.168.1.2 remote-as 40000
  neighbor 192.168.3.2 remote-as 50000
  neighbor 172.16.1.2 activate
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

An EIGRP routing process is then configured and BGP routes are redistributed into EIGRP with a route map filtering the routes:

```
router eigrp 100
 redistribute bgp 45000 metric 10000 100 255 1 1500 route-map bgp-to-eigrp
 no auto-summary
 exit
```

If you later decide to remove the route map, you will use the **no** form of the **route-map** command. Almost every configuration command has a **no** form, and the **no** form generally disables a function. However, in this configuration example, if you disable only the route map, the route redistribution will continue, but without the filtering or matching from the route map. Redistribution without the route map may cause unexpected behavior in your network. When you remove an access list or route map, you must also review the commands that referenced that access list or route map to consider whether the command will give you the behavior you intended.

The following configuration will remove both the route map and the redistribution:

```
configure terminal
 no route-map bgp-to-eigrp
 router eigrp 100
  no redistribute bgp 45000
  end
```

For details on configuring the removal of BGP CLI configuration, see the “Configuring a Basic BGP Network” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1700	*Assigned Numbers*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3107	*Carrying Label Information in BGP-4*
RFC 4271	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4893	*BGP Support for Four-Octet AS Number Space*
RFC 5396	*Textual Representation of Autonomous System (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

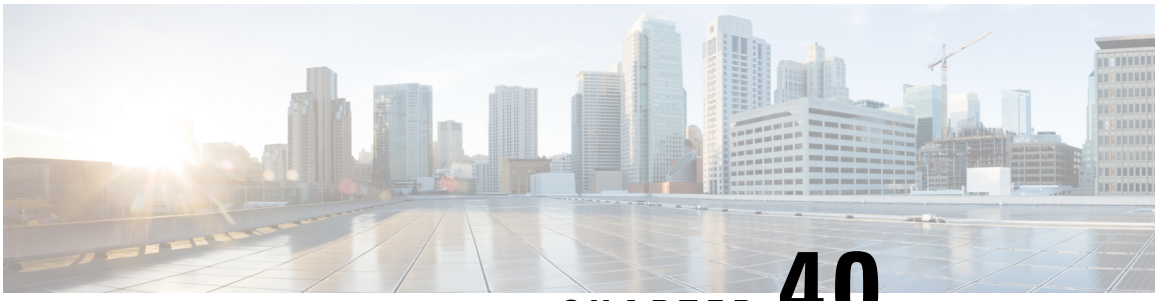
Feature Information for Cisco BGP Overview

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 52: Feature Information for Cisco BGP Overview

Feature Name	Releases	Feature Information
Multiprotocol BGP	Cisco IOS XE 3.1.0SG	Cisco IOS software supports multiprotocol BGP extensions as defined in RFC 2858, *Multiprotocol Extensions for BGP-4* . The extensions introduced in this RFC allow BGP to carry routing information for multiple network layer protocols including CLNS, IPv4, IPv6, and VPNv4. These extensions are backward compatible to enable routers that do not support multiprotocol extensions to communicate with those routers that do support multiprotocol extensions. Multiprotocol BGP carries routing information for multiple network layer protocols and IP multicast routes.



CHAPTER 40

BGP 4

BGP is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems).

Information About BGP 4

BGP Version 4 Functional Overview

BGP is an interdomain routing protocol designed to provide loop-free routing links between organizations. BGP is designed to run over a reliable transport protocol; it uses TCP (port 179) as the transport protocol because TCP is a connection-oriented protocol. The destination TCP port is assigned 179, and the local port is assigned a random port number. Cisco software supports BGP version 4 and it is this version that has been used by Internet service providers (ISPs) to help build the Internet. RFC 1771 introduced and discussed a number of new BGP features to allow the protocol to scale for Internet use. RFC 2858 introduced multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families, including IPv4, IPv6, and CLNS.

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. Although BGP is referred to as an exterior gateway protocol (EGP), many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

BGP uses a path-vector routing algorithm to exchange network reachability information with other BGP-speaking networking devices. Network reachability information is exchanged between BGP peers in routing updates. Network reachability information contains the network number, path-specific attributes, and the list of autonomous system numbers that a route must transit to reach a destination network. This list is contained in the AS-path attribute. BGP prevents routing loops by rejecting any routing update that contains the local autonomous system number because this indicates that the route has already traveled through that

autonomous system and a loop would therefore be created. The BGP path-vector routing algorithm is a combination of the distance-vector routing algorithm and the AS-path loop detection.

BGP selects a single path, by default, as the best path to a destination host or network. The best path selection algorithm analyzes path attributes to determine which route is installed as the best path in the BGP routing table. Each path carries well-known mandatory, well-known discretionary, and optional transitive attributes that are used in BGP best path analysis. Cisco software provides the ability to influence BGP path selection by altering some of these attributes using the command-line interface (CLI.) BGP path selection can also be influenced through standard BGP policy configuration. For more details about using BGP to influence path selection and configuring BGP policies to filter traffic, see the "BGP 4 Prefix Filter and Inbound Route Maps" module and the "BGP Prefix-Based Outbound Route Filtering" module.

BGP uses the best-path selection algorithm to find a set of equally good routes. These routes are the potential multipaths. In Cisco IOS Release 12.2(33)SRD and later releases, when there are more equally good multipaths available than the maximum permitted number, the oldest paths are selected as multipaths.

BGP can be used to help manage complex internal networks by interfacing with Interior Gateway Protocols (IGPs). Internal BGP can help with issues such as scaling the existing IGPs to match the traffic demands while maintaining network efficiency.

Note BGP requires more configuration than other routing protocols and the effects of any configuration changes must be fully understood. Incorrect configuration can create routing loops and negatively impact normal network operation.

BGP Router ID

BGP uses a router ID to identify BGP-speaking peers. The BGP router ID is a 32-bit value that is often represented by an IPv4 address. By default, the Cisco software sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the device, the software chooses the highest IPv4 address configured on a physical interface of the device to represent the BGP router ID. The BGP router ID must be unique to the BGP peers in a network.

BGP-Speaker and Peer Relationships

A BGP-speaking device does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking devices. A peer device is a BGP-speaking device that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a neighbor, but because this can imply the idea that the BGP devices are directly connected with no other device in between, the term *neighbor* will be avoided whenever possible in this document. A BGP speaker is the local device, and a peer is any other BGP-speaking network device.

When a TCP connection is established between peers, each BGP peer initially exchanges all its routes—the complete BGP routing table—with the other peer. After this initial exchange, only incremental updates are sent when there has been a topology change in the network, or when a routing policy has been implemented or modified. In the periods of inactivity between these updates, peers exchange special messages called keepalives.

A BGP autonomous system is a network that is controlled by a single technical administration entity. Peer devices are called external peers when they are in different autonomous systems and internal peers when they

are in the same autonomous system. Usually, external peers are adjacent and share a subnet; internal peers may be anywhere in the same autonomous system.

BGP Peer Session Establishment

When a BGP routing process establishes a peering session with a peer, it goes through the following state changes:

- Idle—The initial state that the BGP routing process enters when the routing process is enabled or when the device is reset. In this state, the device waits for a start event, such as a peering configuration with a remote peer. After the device receives a TCP connection request from a remote peer, the device initiates another start event to wait for a timer before starting a TCP connection to a remote peer. If the device is reset, the peer is reset and the BGP routing process returns to the Idle state.
- Connect—The BGP routing process detects that a peer is trying to establish a TCP session with the local BGP speaker.
- Active—In this state, the BGP routing process tries to establish a TCP session with a peer device using the ConnectRetry timer. Start events are ignored while the BGP routing process is in the Active state. If the BGP routing process is reconfigured or if an error occurs, the BGP routing process will release system resources and return to an Idle state.
- OpenSent—The TCP connection is established, and the BGP routing process sends an OPEN message to the remote peer, and transitions to the OpenSent state. The BGP routing process can receive other OPEN messages in this state. If the connection fails, the BGP routing process transitions to the Active state.
- OpenReceive—The BGP routing process receives the OPEN message from the remote peer and waits for an initial keepalive message from the remote peer. When a keepalive message is received, the BGP routing process transitions to the Established state. If a notification message is received, the BGP routing process transitions to the Idle state. If an error or configuration change occurs that affects the peering session, the BGP routing process sends a notification message with the Finite State Machine (FSM) error code and then transitions to the Idle state.
- Established—The initial keepalive is received from the remote peer. Peering is now established with the remote neighbor and the BGP routing process starts exchanging update message with the remote peer. The hold timer restarts when an update or keepalive message is received. If the BGP process receives an error notification, it will transition to the Idle state.

BGP Session Reset

Whenever the routing policy changes due to a configuration change, BGP peering sessions must be reset by using the **clear ip bgp** command. Cisco software supports the following three mechanisms to reset BGP peering sessions:

- Hard reset—A hard reset tears down the specified peering sessions including the TCP connection and deletes routes coming from the specified peer.
- Soft reset—A soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions. Soft reconfiguration uses stored update information, at the cost of additional memory for storing the updates, to allow you to apply new BGP policy without disrupting the network. Soft reconfiguration can be configured for inbound or outbound sessions.

- Dynamic inbound soft reset—The route refresh capability, as defined in RFC 2918, allows the local device to reset inbound routing tables dynamically by exchanging route refresh requests to supporting peers. The route refresh capability does not store update information locally for nondisruptive policy changes. It instead relies on dynamic exchange with supporting peers. Route refresh must first be advertised through BGP capability negotiation between peers. All BGP devices must support the route refresh capability. To determine if a BGP device supports this capability, use the **show ip bgp neighbors** command. The following message is displayed in the output when the device supports the route refresh capability:

```
Received route refresh capability from peer.
```

The **bgp soft-reconfig-backup** command was introduced to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command.

BGP Route Aggregation

BGP peers store and exchange routing information and the amount of routing information increases as more BGP speakers are configured. The use of route aggregation reduces the amount of information involved. Aggregation is the process of combining the attributes of several different routes so that only a single route is advertised. Aggregate prefixes use the classless interdomain routing (CIDR) principle to combine contiguous networks into one classless set of IP addresses that can be summarized in routing tables. Fewer routes now need to be advertised.

Two methods are available in BGP to implement route aggregation. You can redistribute an aggregated route into BGP or you can use a form of conditional aggregation. Basic route redistribution involves creating an aggregate route and then redistributing the routes into BGP. Conditional aggregation involves creating an aggregate route and then advertising or suppressing the advertising of certain routes on the basis of route maps, autonomous system set path (AS-SET) information, or summary information.

The **bgp suppress-inactive** command configures BGP to not advertise inactive routes to any BGP peer. A BGP routing process can advertise routes that are not installed in the routing information database (RIB) to BGP peers by default. A route that is not installed into the RIB is an inactive route. Inactive route advertisement can occur, for example, when routes are advertised through common route aggregation. Inactive route advertisements can be suppressed to provide more consistent data forwarding.

BGP Route Aggregation Generating AS_SET Information

AS_SET information can be generated when BGP routes are aggregated using the **aggregate-address** command. The path advertised for such a route is an AS_SET consisting of all the elements, including the communities, contained in all the paths that are being summarized. If the AS_PATHs to be aggregated are identical, only the AS_PATH is advertised. The ATOMIC-AGGREGATE attribute, set by default for the **aggregate-address** command, is not added to the AS_SET.

Routing Policy Change Management

Routing policies for a peer include all the configurations for elements such as a route map, distribute list, prefix list, and filter list that may impact inbound or outbound routing table updates. Whenever there is a change in the routing policy, the BGP session must be soft-cleared, or soft-reset, for the new policy to take

effect. Performing inbound reset enables the new inbound policy configured on the device to take effect. Performing outbound reset causes the new local outbound policy configured on the device to take effect without resetting the BGP session. As a new set of updates is sent during outbound policy reset, a new inbound policy of the neighbor can also take effect. This means that after changing inbound policy, you must do an inbound reset on the local device or an outbound reset on the peer device. Outbound policy changes require an outbound reset on the local device or an inbound reset on the peer device.

There are two types of reset: hard reset and soft reset. The table below lists their advantages and disadvantages.

Table 53: Advantages and Disadvantages of Hard and Soft Resets

Type of Reset	Advantages	Disadvantages
Hard reset	No memory overhead.	The prefixes in the BGP, IP, and Forwarding Information Base (FIB) tables provided by the neighbor are lost. A hard reset is not recommended.
Outbound soft reset	No configuration, and no storing of routing table updates.	Does not reset inbound routing table updates.
Dynamic inbound soft reset	Does not clear the BGP session and cache. Does not require storing of routing table updates, and has no memory overhead.	Both BGP devices must support the route refresh capability. **Note** Does not reset outbound routing table updates.
Configured inbound soft reset (uses the **neighbor soft-reconfiguration** router configuration command)	Can be used when both BGP devices do not support the automatic route refresh capability. The **bgp soft-reconfig-backup** command was introduced to configure inbound soft reconfiguration for peers that do not support the route refresh capability.	Requires preconfiguration. Stores all received (inbound) routing policy updates without modification; is memory-intensive. Recommended only when absolutely necessary, such as when both BGP devices do not support the automatic route refresh capability. **Note** Does not reset outbound routing table updates.

Once you have defined two devices to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or if you make a similar configuration change, you must reset BGP connections in order for the configuration change to take effect.

A soft reset updates the routing table for inbound and outbound routing updates. Cisco software supports soft reset without any prior configuration. This soft reset allows the dynamic exchange of route refresh requests and routing information between BGP devices, and allows the subsequent readvertisement of the respective outbound routing table. There are two types of soft reset:

- When soft reset is used to generate inbound updates from a neighbor, it is called dynamic inbound soft reset.
- When soft reset is used to send a new set of updates to a neighbor, it is called outbound soft reset.

To use soft reset without preconfiguration, both BGP peers must support the soft route refresh capability, which is advertised in the OPEN message sent when the peers establish a TCP session.

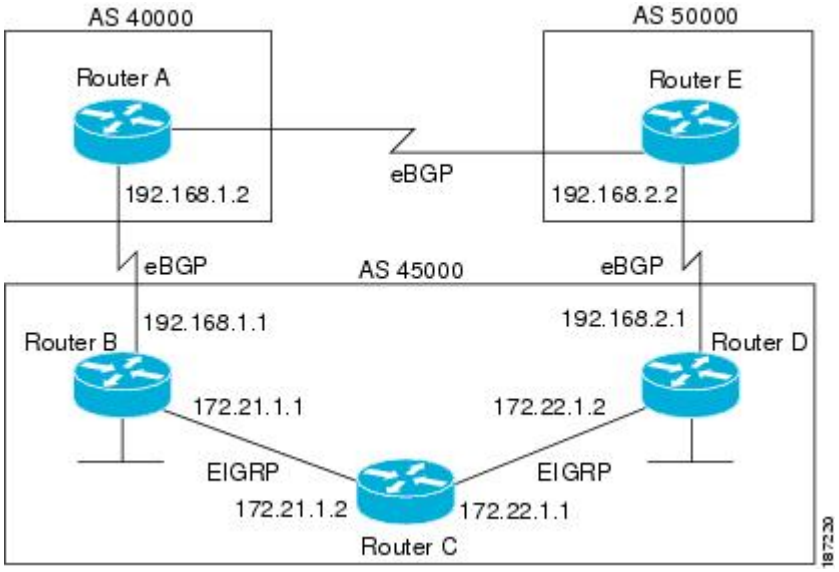
BGP Peer Groups

Often, in a BGP network, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into BGP peer groups to simplify configuration and, more importantly, to make configuration updates more efficient. When you have many peers, this approach is highly recommended.

BGP Backdoor Routes

In a BGP network topology with two border devices using eBGP to communicate to a number of different autonomous systems, using eBGP to communicate between the two border devices may not be the most efficient routing method. In the figure below, Router B as a BGP speaker will receive a route to Router D through eBGP, but this route will traverse at least two autonomous systems. Router B and Router D are also connected through an Enhanced Interior Gateway Routing Protocol (EIGRP) network (any IGP can be used here), and this route has a shorter path. EIGRP routes, however, have a default administrative distance of 90, and eBGP routes have a default administrative distance of 20, so BGP will prefer the eBGP route. Changing the default administrative distances is not recommended because changing the administrative distance may lead to routing loops. To cause BGP to prefer the EIGRP route, you can use the **network backdoor** command. BGP treats the network specified by the **network backdoor** command as a locally assigned network, except that it does not advertise the specified network in BGP updates. In the figure below, this means that Router B will communicate to Router D using the shorter EIGRP route instead of the longer eBGP route.

Figure 36: BGP Backdoor Route Topology



How to Configure BGP 4

Configuring a basic BGP network consists of a few required tasks and many optional tasks. A BGP routing process must be configured and BGP peers must be configured, preferably using the address family configuration model. If the BGP peers are part of a VPN network, the BGP peers must be configured using the IPv4 VRF address family task.

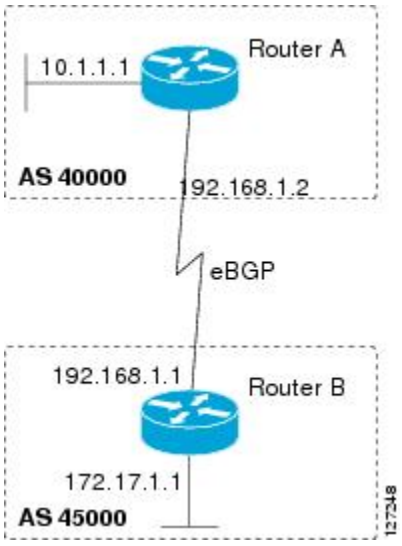
Configuring a BGP Routing Process

Perform this task to configure a BGP routing process. You must perform the required steps at least once to enable BGP. The optional steps here allow you to configure additional features in your BGP network. Several of the features, such as logging neighbor resets and immediate reset of a peer when its link goes down, are enabled by default but are presented here to enhance your understanding of how your BGP network operates.

Note A device that runs Cisco software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple concurrent BGP address family and subaddress family configurations.

The configuration in this task is done at Router A in the figure below and would need to be repeated with appropriate changes to the IP addresses (for example, at Router B) to fully achieve a BGP process between the two devices. No address family is configured here for the BGP routing process, so routing information for the IPv4 unicast address family is advertised by default.

Figure 37: BGP Topology with Two Autonomous Systems



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*

4. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
5. **bgp router-id** *ip-address*
6. **timers bgp** *keepalive holdtime*
7. **bgp fast-external-fallover**
8. **bgp log-neighbor-changes**
9. **end**
10. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Configures a BGP routing process, and enters router configuration mode for the specified routing process. • Use the *autonomous-system-number* argument to specify an integer, from 0 and 65534, that identifies the device to other BGP speakers.
Step 4	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router)# network 10.1.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols, the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 5	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id 10.1.1.99`	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. • Use the *ip-address* argument to specify a unique router ID within the network. **Note** Configuring a router ID using the **bgp router-id** command resets all active BGP peering sessions.
Step 6	**timers bgp** *keepalive holdtime* **Example:** `Device(config-router)# timers bgp 70 120`	(Optional) Sets BGP network timers. • Use the *keepalive* argument to specify the frequency, in seconds, with which the software sends keepalive

	Command or Action	Purpose
		messages to its BGP peer. By default, the keepalive timer is set to 60 seconds. • Use the *holdtime* argument to specify the interval, in seconds, after which the software, having not received a keepalive message, declares a BGP peer dead. By default, the holdtime timer is set to 180 seconds.
Step 7	**bgp fast-external-fallover** **Example:** `Device(config-router)# bgp fast-external-fallover`	(Optional) Enables the automatic resetting of BGP sessions. • By default, the BGP sessions of any directly adjacent external peers are reset if the link used to reach them goes down.
Step 8	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets. • Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 10	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output from the **show ip bgp** command shows the BGP routing table for Router A in the figure above after this task has been configured on Router A. You can see an entry for the network 10.1.1.0 that is local to this autonomous system.

```
BGP table version is 12, local router ID is 10.1.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
```

Troubleshooting Tips

Use the **ping** command to check basic network connectivity between the BGP routers.

Configuring a BGP Peer

Perform this task to configure BGP between two IPv4 devices (peers). The address family configured here is the default IPv4 unicast address family, and the configuration is done at Router A in the figure above. Remember to perform this task for any neighboring devices that are to be BGP peers.

Before you begin

Before you perform this task, perform the "Configuring a BGP Routing Process" task.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, such as IPv6 prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **neighbor** *ip-address* **activate**
7. **end**
8. **show ip bgp** [*network*] [*network-mask*]
9. **show ip bgp neighbors** [*neighbor-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:**	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 40000`	
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the device is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local device.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 9	**show ip bgp neighbors** [*neighbor-address*] **Example:** `Device(config-router-af)# show ip bgp neighbors 192.168.2.2`	(Optional) Displays information about the TCP and BGP connections to neighbors. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output from the **show ip bgp** command shows the BGP routing table for Router A in the figure above after this task has been configured on Router A and Router B. You can now see an entry for the network 172.17.1.0 in autonomous system 45000.

```
BGP table version is 13, local router ID is 10.1.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*> 172.17.1.0/24    192.168.1.1              0             0 45000 i
```

The following sample output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.1.1 of Router A in the figure above after this task has been configured on Router A:

```
BGP neighbor is 192.168.1.1,  remote AS 45000, external link
  BGP version 4, remote router ID 172.17.1.99
  BGP state = Established, up for 00:06:55
  Last read 00:00:15, last write 00:00:15, hold time is 120, keepalive intervals
  Configured hold time is 120,keepalive interval is 70 seconds, Minimum holdtims
  Neighbor capabilities:
    Route refresh: advertised and received (old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            13         13
    Route Refresh:          0          0
    Total:                 15         16
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 13, neighbor version 13/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:             1          1 (Consumes 52 bytes)
    Prefixes Total:               1          1
    Implicit Withdraw:            0          0
    Explicit Withdraw:            0          0
    Used as bestpath:           n/a          1
    Used as multipath:          n/a          0
                                 Outbound    Inbound
  Local Policy Denied Prefixes:  --------    -------
    AS_PATH loop:                    n/a          1
    Bestpath from this peer:           1        n/a
    Total:                             1          1
  Number of NLRIs in the update sent: max 0, min 0
  Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
```

```
Local host: 192.168.1.2, Local port: 179
Foreign host: 192.168.1.1, Foreign port: 37725
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x12F4F2C):
Timer          Starts    Wakeups            Next
Retrans            14          0             0x0
TimeWait            0          0             0x0
AckHold            13          8             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss:  165379618  snduna:  165379963  sndnxt:  165379963     sndwnd:  16040
irs: 3127821601  rcvnxt: 3127821993  rcvwnd:      15993  delrcvwnd:    391
SRTT: 254 ms, RTTO: 619 ms, RTV: 365 ms, KRTT: 0 ms
minRTT: 12 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
IP Precedence value : 6
Datagrams (max data segment is 1460 bytes):
Rcvd: 20 (out of order: 0), with data: 15, total data bytes: 391
Sent: 22 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 04
```

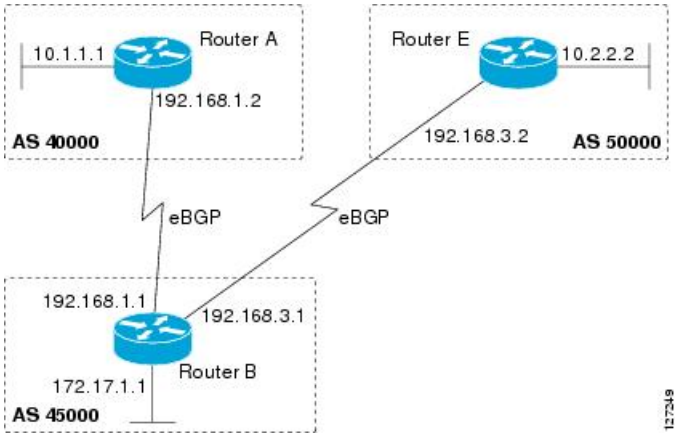
Troubleshooting Tips

Use the **ping** command to verify basic network connectivity between the BGP devices.

Configuring a BGP Peer for the IPv4 VRF Address Family

Perform this optional task to configure BGP between two IPv4 devices (peers) that must exchange IPv4 VRF information because they exist in a VPN. The address family configured here is the IPv4 VRF address family, and the configuration is done at Router B in the figure below with the neighbor 192.168.3.2 at Router E in autonomous system 50000. Remember to perform this task for any neighboring devices that are to be BGP IPv4 VRF address family peers.

Figure 38: BGP Topology for IPv4 VRF Address Family



Before you begin

Before you perform this task, perform the “Configuring a BGP Routing Process” task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
6. **exit**
7. **ip vrf** *vrf-name*
8. **rd** *route-distinguisher*
9. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
10. **exit**
11. **router bgp** *autonomous-system-number*
12. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*] [**restart** *restart-interval*] [**warning-only**]
15. **neighbor** *ip-address* **activate**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vpn1`	Associates a VPN VRF instance with an interface or subinterface.
Step 5	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** `Device(config-if)# ip address 192.168.3.1 255.255.255.0`	Sets an IP address for an interface.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vpn1`	Configures a VRF routing table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF.
Step 8	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 45000:5`	Creates routing and forwarding tables and specifies the default route distinguisher for a VPN. • Use the *route-distinguisher* argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.
Step 9	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target both 45000:100`	Creates a route target extended community for a VRF. • Use the **import** keyword to import routing information from the target VPN extended community. • Use the **export** keyword to export routing information to the target VPN extended community. • Use the **both** keyword to import both import and export routing information to the target VPN extended community. • Use the *route-target-ext-community* argument to add the route target extended community attributes to the VRF's list of import, export, or both (import and export) route target extended communities.
Step 10	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 11	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 12	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:**	Specifies the IPv4 address family and enters address family configuration mode. • Use the **unicast** keyword to specify the IPv4 unicast address family. By default, the device is placed in

	Command or Action	Purpose
	`Device(config-router)# address-family ipv4 vrf vpn1`	configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.3.2 remote-as 50000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 14	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*] [**restart** *restart-interval*] [**warning-only**] **Example:** `Device(config-router-af)# neighbor 192.168.3.2 maximum-prefix 10000 warning-only`	Controls how many prefixes can be received from a neighbor. • Use the *maximum* argument to specify the maximum number of prefixes allowed from the specified neighbor. The number of prefixes that can be configured is limited only by the available system resources on a device. • Use the *threshold* argument to specify an integer representing a percentage of the maximum prefix limit at which the device starts to generate a warning message. • Use the **warning-only** keyword to allow the device to generate a log message when the maximum prefix limit is exceeded, instead of terminating the peering session.
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.3.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local device.
Step 16	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

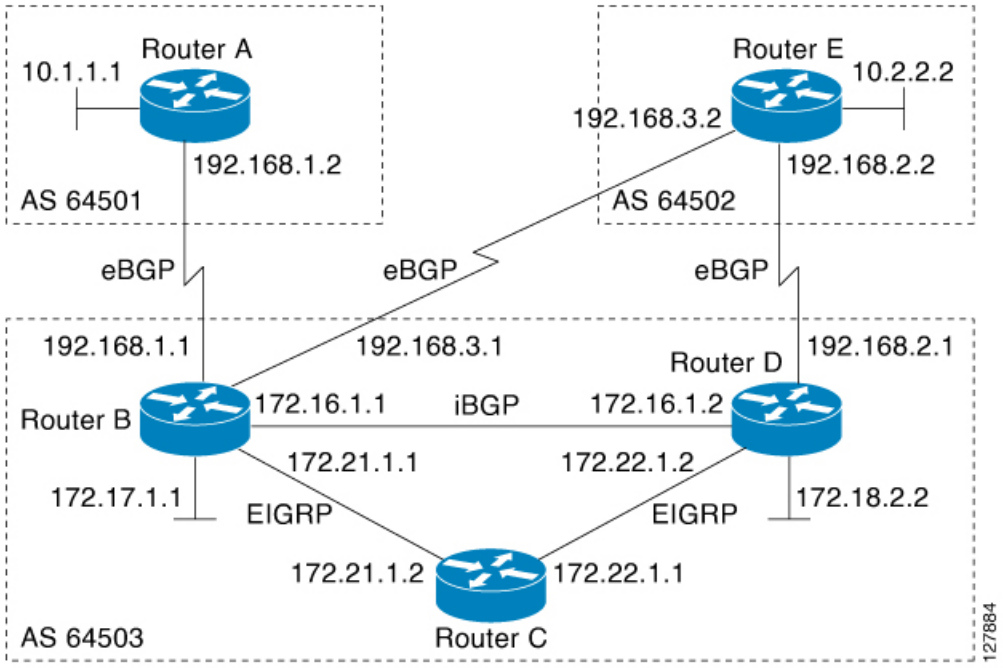
Use the **ping vrf** command to verify basic network connectivity between the BGP devices, and use the **show ip vrf** command to verify that the VRF instance has been created.

Customizing a BGP Peer

Perform this task to customize your BGP peers. Although many of the steps in this task are optional, this task demonstrates how the neighbor and address family configuration command relationships work. Using the example of the IPv4 multicast address family, neighbor address family-independent commands are configured before the IPv4 multicast address family is configured. Commands that are address family-dependent are then configured and the **exit address-family** command is shown. An optional step shows how to disable a neighbor.

The configuration in this task is done at Router B in the figure below and would need to be repeated with appropriate changes to the IP addresses, for example, at Router E to fully configure a BGP process between the two devices.

Figure 39: BGP Peer Topology



Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, such as IPv6 prefixes.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **description** *text*
7. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
8. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
9. **neighbor** {*ip-address* | *peer-group-name*} **activate**
10. **neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds*
11. **neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]
12. **exit-address-family**
13. **neighbor** {*ip-address* | *peer-group-name*} **shutdown**
14. **end**
15. **show ip bgp ipv4 multicast** [*command*]
16. **show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received prefix-filter**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast** router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.3.2 remote-as 50000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **description** *text* **Example:** `Device(config-router)# neighbor 192.168.3.2 description finance`	(Optional) Associates a text description with the specified neighbor.
Step 7	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 multicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the device is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 8	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.3.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds* **Example:**	(Optional) Sets the minimum interval between the sending of BGP routing updates.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.3.2 advertisement-interval 25`	
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*] **Example:** `Device(config-router-af)# neighbor 192.168.3.2 default-originate`	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 12	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 13	**neighbor** {*ip-address* | *peer-group-name*} **shutdown** **Example:** `Device(config-router)# neighbor 192.168.3.2 shutdown`	(Optional) Disables a BGP peer or peer group. **Note** If you perform this step you will not be able to run either of the subsequent **show** command steps because you have disabled the neighbor.
Step 14	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 15	**show ip bgp ipv4 multicast** [*command*] **Example:** `Device# show ip bgp ipv4 multicast`	(Optional) Displays IPv4 multicast database-related information. • Use the *command* argument to specify any multiprotocol BGP command that is supported. To see the supported commands, use the ? prompt on the CLI.
Step 16	**show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received prefix-filter**] **Example:** `Device# show ip bgp neighbors 192.168.3.2`	(Optional) Displays information about the TCP and BGP connections to neighbors.

Examples

The following sample output from the **show ip bgp ipv4 multicast** command shows BGP IPv4 multicast information for Router B in the figure above after this task has been configured on Router B and Router E. Note that the networks local to each device that were configured under IPv4 multicast address family appear in the output table.

```
BGP table version is 3, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.3.2              0             0 50000 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

The following partial sample output from the **show ip bgp neighbors** command for neighbor 192.168.3.2 shows general BGP information and specific BGP IPv4 multicast address family information about the neighbor. The command was entered on Router B in the figure above after this task had been configured on Router B and Router E.

```
BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Description: finance
  BGP version 4, remote router ID 10.2.2.99
  BGP state = Established, up for 01:48:27
  Last read 00:00:26, last write 00:00:26, hold time is 120, keepalive intervals
  Configured hold time is 120,keepalive interval is 70 seconds, Minimum holdtims
  Neighbor capabilities:
    Route refresh: advertised and received (old & new)
    Address family IPv4 Unicast: advertised
    Address family IPv4 Multicast: advertised and received
!
 For address family: IPv4 Multicast
  BGP table version 3, neighbor version 3/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
    Uses NEXT_HOP attribute for MBGP NLRIs
                                 Sent       Rcvd
  Prefix activity:               ----       ----
    Prefixes Current:               1          1 (Consumes 48 bytes)
    Prefixes Total:                 1          1
    Implicit Withdraw:              0          0
    Explicit Withdraw:              0          0
    Used as bestpath:             n/a          1
    Used as multipath:            n/a          0
                                   Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    Bestpath from this peer:                1        n/a
    Total:                                  1          0
  Number of NLRIs in the update sent: max 0, min 0
  Minimum time between advertisement runs is 25 seconds
  Connections established 8; dropped 7
  Last reset 01:48:54, due to User reset
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 192.168.3.1, Local port: 13172
Foreign host: 192.168.3.2, Foreign port: 179
!
```

Removing BGP Configuration Commands Using a Redistribution

BGP CLI configuration can become quite complex even in smaller BGP networks. If you need to remove any CLI configuration, you must consider all the implications of removing the CLI. Analyze the current running configuration to determine the current BGP neighbor relationships, any address family considerations, and even other routing protocols that are configured. Many BGP CLI commands affect other parts of the CLI configuration.

Perform this task to remove all the BGP configuration commands used in a redistribution of BGP routes into EIGRP. A route map can be used to match and set parameters or to filter the redistributed routes to ensure that routing loops are not created when these routes are subsequently advertised by EIGRP. When removing BGP configuration commands you must remember to remove or disable all the related commands. In this example, if the **route-map** command is omitted, then the redistribution will still occur and possibly with unexpected results as the route map filtering has been removed. Omitting just the **redistribute** command would mean that the route map is not applied, but it would leave unused commands in the running configuration.

For more details on BGP CLI removal, see the "BGP CLI Removal Considerations" concept in the "Cisco BGP Overview" module.

To view the redistribution configuration before and after the CLI removal, see the "Examples: Removing BGP Configuration Commands Using a Redistribution Example" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no route-map** *map-name*
4. **router eigrp** *autonomous-system-number*
5. **no redistribute** *protocol* [*as-number*]
6. **end**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no route-map** *map-name* **Example:** `Device(config)# no route-map bgp-to-eigrp`	Removes a route map from the running configuration. • In this example, a route map named bgp-to-eigrp is removed from the configuration.
Step 4	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 100`	Enters router configuration mode for the specified routing process.
Step 5	**no redistribute** *protocol* [*as-number*] **Example:**	Disables the redistribution of routes from one routing domain into another routing domain.

	Command or Action	Purpose
	`Device(config-router)# no redistribute bgp 45000`	• In this example, the configuration of the redistribution of BGP routes into the EIGRP routing process is removed from the running configuration. **Note** If a route map was included in the original **redistribute** command configuration, remember to remove the **route-map** command configuration as in Step 3 in this example task. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	(Optional) Displays the current running configuration on the router. • Use this command to verify that the **redistribute** and **route-map** commands are removed from the router configuration.

Monitoring and Maintaining Basic BGP

The tasks in this section are concerned with the resetting and display of information about basic BGP processes and peer relationships. Once you have defined two devices to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you may have to reset BGP connections for the configuration change to take effect.

Configuring Inbound Soft Reconfiguration When Route Refresh Capability Is Missing

Perform this task to configure inbound soft reconfiguration using the **bgp soft-reconfig-backup** command for BGP peers that do not support the route refresh capability. BGP peers that support the route refresh capability are unaffected by the configuration of this command. Note that the memory requirements for storing the inbound update information can become quite large.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp log-neighbor-changes**
5. **bgp soft-reconfig-backup**
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*

7. **neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [**inbound**]
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.
10. **exit**
11. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address*
13. **end**
14. **show ip bgp neighbors** [*neighbor-address*]
15. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**bgp soft-reconfig-backup** **Example:** `Device(config-router)# bgp soft-reconfig-backup`	Configures a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability. • This command is used to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:**	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.

	Command or Action	Purpose
	`Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [**inbound**] **Example:** `Device(config-router)# neighbor 192.168.1.2 soft-reconfiguration inbound`	Configures the Cisco software to start storing updates. • All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is done later, the stored information will be used to generate a new set of inbound updates.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router)# neighbor 192.168.1.2 route-map LOCAL in`	Applies a route map to incoming or outgoing routes. • In this example, the route map named LOCAL will be applied to incoming routes.
Step 9	Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.	—
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 11	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map LOCAL permit 10`	Configures a route map and enters route-map configuration mode. • In this example, a route map named LOCAL is created.
Step 12	**set ip next-hop** *ip-address* **Example:** `Device(config-route-map)# set ip next-hop 192.168.1.144`	Specifies where output packets that pass a match clause of a route map for policy routing. • In this example, the ip address is set to 192.168.1.144.
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	**show ip bgp neighbors** [*neighbor-address*] **Example:** `Device# show ip bgp neighbors 192.168.1.2`	(Optional) Displays information about the TCP and BGP connections to neighbors. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

	Command or Action	Purpose
Step 15	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following partial output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.2.1. This peer supports route refresh.

```
BGP neighbor is 192.168.1.2,  remote AS 40000, external link
 Neighbor capabilities:
   Route refresh: advertised and received(new)
```

The following partial output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.3.2. This peer does not support route refresh so the soft-reconfig inbound paths for BGP peer 192.168.3.2 will be stored because there is no other way to update any inbound policy updates.

```
BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Neighbor capabilities:
   Route refresh: advertised
```

The following sample output from the **show ip bgp** command shows the entry for the network 172.17.1.0. Both BGP peers are advertising 172.17.1.0/24, but only the received-only path is stored for 192.168.3.2.

```
BGP routing table entry for 172.17.1.0/24, version 11
Paths: (3 available, best #3, table Default-IP-Routing-Table, RIB-failure(4))
Flag: 0x820
 Advertised to update-groups:
    1
 50000
   192.168.3.2 from 192.168.3.2 (172.17.1.0)
     Origin incomplete, metric 0, localpref 200, valid, external
 50000, (received-only)
   192.168.3.2 from 192.168.3.2 (172.17.1.0)
     Origin incomplete, metric 0, localpref 100, valid, external
 40000
   192.168.1.2 from 192.168.1.2 (172.16.1.0)
     Origin incomplete, metric 0, localpref 200, valid, external, best
```

Resetting and Displaying Basic BGP Information

Perform this task to reset and display information about basic BGP processes and peer relationships.

SUMMARY STEPS

1. **enable**
2. **clear ip bgp** {***** | *autonomous-system-number* | *neighbor-address*} [**soft** [**in** | **out**]]

3. **show ip bgp** [*network-address*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter prefixes** *mask-length*]
4. **show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*]
5. **show ip bgp paths**
6. **show ip bgp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp** {***** | *autonomous-system-number* | *neighbor-address*} [**soft** [**in** | **out**]] **Example:** `Device# clear ip bgp *`	Clears and resets BGP neighbor sessions: • In the example provided, all BGP neighbor sessions are cleared and reset.
Step 3	**show ip bgp** [*network-address*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter prefixes** *mask-length*] **Example:** `Device# show ip bgp 10.1.1.0 255.255.255.0`	Displays all the entries in the BGP routing table: • In the example provided, the BGP routing table information for the 10.1.1.0 network is displayed.
Step 4	**show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*] **Example:** `Device# show ip bgp neighbors 192.168.3.2 advertised-routes`	Displays information about the TCP and BGP connections to neighbors. • In the example provided, the routes advertised from the device to BGP neighbor 192.168.3.2 on another device are displayed.
Step 5	**show ip bgp paths** **Example:** `Device# show ip bgp paths`	Displays information about all the BGP paths in the database.
Step 6	**show ip bgp summary** **Example:** `Device# show ip bgp summary`	Displays information about the status of all BGP connections.

Aggregating Route Prefixes Using BGP

BGP peers exchange information about local networks, but this can quickly lead to large BGP routing tables. CIDR enables the creation of aggregate routes (or *supernets*) to minimize the size of routing tables. Smaller BGP routing tables can reduce the convergence time of the network and improve network performance. Aggregated routes can be configured and advertised using BGP. Some aggregations advertise only summary routes and other methods of aggregating routes allow more specific routes to be forwarded. Aggregation applies only to routes that exist in the BGP routing table. An aggregated route is forwarded if at least one more specific route of the aggregation exists in the BGP routing table. Perform one of the following tasks to aggregate routes within BGP:

Redistributing a Static Aggregate Route into BGP

Use this task to redistribute a static aggregate route into BPG. A static aggregate route is configured and then redistributed into the BGP routing table. The static route must be configured to point to interface null 0 and the prefix should be a superset of known BGP routes. When a device receives a BGP packet, it will use the more specific BGP routes. If the route is not found in the BGP routing table, then the packet will be forwarded to null 0 and discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
4. **router bgp** *autonomous-system-number*
5. **redistribute static**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*] **Example:** `Device(config)# ip route 172.0.0.0 255.0.0.0 null 0`	Creates a static route.

	Command or Action	Purpose
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 5	**redistribute static** **Example:** `Device(config-router)# redistribute static`	Redistributes routes into the BGP routing table.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Conditional Aggregate Routes Using BGP

Use this task to create an aggregate route entry in the BGP routing table when at least one specific route falls into the specified range. The aggregate route is advertised as originating from your autonomous system. For more information, see the "BGP Route Aggregation Generating AS_SET Information" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **aggregate-address** *address mask* [**as-set**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	**aggregate-address** *address mask* [**as-set**] **Example:** `Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 as-set`	Creates an aggregate entry in a BGP routing table. • A specified route must exist in the BGP table. • Use the **aggregate-address** command with no keywords to create an aggregate entry if any more-specific BGP routes are available that fall in the specified range. • Use the **as-set** keyword to specify that the path advertised for this route is an AS_SET. Do not use the **as-set** keyword when aggregating many paths because this route is withdrawn and updated every time the reachability information for the aggregated route changes. **Note** Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Suppressing and Unsuppressing the Advertisement of Aggregated Routes Using BGP

Use this task to create an aggregate route, suppress the advertisement of routes using BGP, and subsequently unsuppress the advertisement of routes. Routes that are suppressed are not advertised to any neighbors, but it is possible to unsuppress routes that were previously suppressed to specific neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. Do one of the following:
   - **aggregate-address** *address mask* [**summary-only**]
   - **aggregate-address** *address mask* [**suppress-map** *map-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **unsuppress-map** *map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.

Step 5	Do one of the following:
	• **aggregate-address** *address mask* [**summary-only**]
	• **aggregate-address** *address mask* [**suppress-map** *map-name*]
	Example:
	`Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 summary-only`
	Example:
	`Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 suppress-map map1`
		Creates an aggregate route.
		• Use the optional **summary-only** keyword to create the aggregate route (for example, 10.*.*.*) and also suppresses advertisements of more-specific routes to all neighbors.
		• Use the optional **suppress-map** keyword to create the aggregate route but suppress advertisement of specified routes. Routes that are suppressed are not advertised to any neighbors. You can use the **match** clauses of route maps to selectively suppress some more-specific routes of the aggregate and leave others unsuppressed. IP access lists and autonomous system path access lists **match** clauses are supported.
		Note Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Step 6	**neighbor** {*ip-address* | *peer-group-name*} **unsuppress-map** *map-name* **Example:** `Device(config-router)# neighbor 192.168.1.2 unsuppress map1`	(Optional) Selectively advertises routes previously suppressed by the **aggregate-address** command. • In this example, the routes previously suppressed in Step 5 are advertised to neighbor 192.168.1.2.
Step 7	**end** **Example:**	Exits router configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router)# end`	

Conditionally Advertising BGP Routes

Perform this task to conditionally advertise selected BGP routes. The routes or prefixes that will be conditionally advertised are defined in two route maps: an advertise map and either an exist map or nonexist map. The route map associated with the exist map or nonexist map specifies the prefix that the BGP speaker will track. The route map associated with the advertise map specifies the prefix that will be advertised to the specified neighbor when the condition is met.

- If a prefix is found to be present in the exist map by the BGP speaker, the prefix specified by the advertise map is advertised.
- If a prefix is found not to be present in the nonexist map by the BGP speaker, the prefix specified by the advertise map is advertised.

If the condition is not met, the route is withdrawn and conditional advertisement does not occur. All routes that may be dynamically advertised or not advertised must exist in the BGP routing table in order for conditional advertisement to occur. These routes are referenced from an access list or an IP prefix list. Note, when configuring an advertise-map, ensure that BGP attributes are set within the advertise-map and not in a separate route-map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **advertise-map** *map-name* {**exist-map** *map-name* | **non-exist-map** *map-name*}
6. **exit**
7. **route-map** *map-tag* [**permit** | **deny**] [**sequence-number**]
8. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
12. **exit**
13. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
14. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 5	**neighbor** *ip-address* **advertise-map** *map-name* {**exist-map** *map-name* | **non-exist-map** *map-name*} **Example:** `Device(config-router)# neighbor 192.168.1.2 advertise-map map1 exist-map map2`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. • In this example, the prefix (172.17.0.0) matching the ACL in the advertise map (the route map named map1) will be advertised to the neighbor only when a prefix (192.168.50.0) matching the ACL in exist map (the route-map named map2) is in the local BGP table.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 7	**route-map** *map-tag* [**permit** | **deny**] [**sequence-number**] **Example:** `Device(config)# route-map map1 permit 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named map1 is created.
Step 8	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Device(config-route-map)# match ip address 1`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match a prefix permitted by access list 1.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map map2 permit 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named map2 is created.
Step 11	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Device(config-route-map)# match ip address 2`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match a prefix permitted by access list 2.
Step 12	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 13	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Device(config)# access-list 1 permit 172.17.0.0`	Configures a standard access list. • In this example, access list 1 permits advertising of the 172.17.0.0 prefix, depending on other conditions set by the **neighbor advertise-map** command.
Step 14	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Device(config)# access-list 2 permit 192.168.50.0`	Configures a standard access list. • In this example, access list 2 permits the 192.168.50.0 to be the prefix of the exist-map.
Step 15	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Originating BGP Routes

Route aggregation is useful to minimize the size of the BGP table, but there are situations when you want to add more specific prefixes to the BGP table. Route aggregation can hide more specific routes. Using the **network** command as shown in the "Configuring a BGP Routing Process" section originates routes, and the following optional tasks originate BGP routes for the BGP table for different situations.

Advertising a Default Route Using BGP

Perform this task to advertise a default route to BGP peers. The default route is locally originated. A default route can be useful to simplify configuration or to prevent the device from using too many system resources. If the device is peered with an Internet service provider (ISP), the ISP will carry full routing tables, so configuring a default route into the ISP network saves resources at the local device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network / length* | **permit** *network / length*} [**ge** *ge-value*] [**le** *le-value*]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
6. **exit**
7. **router bgp** *autonomous-system-number*
8. **neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network / length*	**permit** *network / length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Device(config)# ip prefix-list DEFAULT permit 10.1.1.0/24`	Configures an IP prefix list. • In this example, prefix list DEFAULT permits advertising of the 10.1.1.0/24. prefix depending on a match set by the **match ip address** command.
Step 4	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] **Example:** `Device(config)# route-map ROUTE`	Configures a route map and enters route map configuration mode. • In this example, a route map named ROUTE is created.

	Command or Action	Purpose
Step 5	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Device(config-route-map)# match ip address prefix-list DEFAULT`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match a prefix permitted by prefix list DEFAULT.
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 7	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*] **Example:** `Device(config-router)# neighbor 192.168.3.2 default-originate`	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Originating BGP Routes Using Backdoor Routes

Use this task to indicate to border devices which networks are reachable using a backdoor route. A backdoor network is treated the same as a local network, except that it is not advertised. For more information, see the BGP Backdoor Routes section.

Before you begin

This task assumes that the IGP (EIGRP, in this example) is already configured for the BGP peers. The configuration is done at Router B in the in the "BGP Backdoor Routes" section, and the BGP peer is Router D.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*

4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **network** *ip-address* **backdoor**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 172.22.1.2 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device. • In this example, the peer is an internal peer as the autonomous system number specified for the peer is the same number specified in Step 3.
Step 5	**network** *ip-address* **backdoor** **Example:** `Device(config-router)# network 172.21.1.0 backdoor`	Indicates a network that is reachable through a backdoor route.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring a BGP Peer Group

This task explains how to configure a BGP peer group. Often, in a BGP speaker, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into peer groups to simplify configuration and, more importantly, to make updating more efficient. When you have many peers, this approach is highly recommended.

The three steps to configure a BGP peer group, described in the following task, are as follows:

- Creating the peer group
- Assigning options to the peer group
- Making neighbors members of the peer group

You can disable a BGP peer or peer group without removing all the configuration information using the **neighbor shutdown** router configuration command.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **peer-group** *peer-group-name*
7. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
8. **neighbor** *peer-group-name* **activate**
9. **neighbor** *ip-address* **peer-group** *peer-group-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *peer-group-name* **peer-group** **Example:**	Creates a BGP peer group.

	Command or Action	Purpose
	`Device(config-router)# neighbor fingroup peer-group`	
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device.
Step 6	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router)# neighbor 192.168.1.1 peer-group fingroup`	Assigns the IP address of a BGP neighbor to a peer group.
Step 7	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 multicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. This is the default. • The **multicast** keyword specifies that IPv4 multicast address prefixes will be exchanged. • The **vrf** keyword and *vrf-name* argument specify that IPv4 VRF instance information will be exchanged.
Step 8	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor fingroup activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local device. **Note** By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. To allow BGP to exchange other address prefix types, such as multicast that is configured in this example, neighbors must also be activated using the **neighbor activate** command.
Step 9	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 peer-group fingroup`	Assigns the IP address of a BGP neighbor to a peer group.
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP 4

Example: Configuring a BGP Process and Customizing Peers

The following example shows the configuration for Router B in the above (in the "Customizing a BGP Peer" section) with a BGP process configured with two neighbor peers (at Router A and at Router E) in separate autonomous systems. IPv4 unicast routes are exchanged with both peers and IPv4 multicast routes are exchanged with the BGP peer at Router E.

Router B

```
router bgp 45000
 bgp router-id 172.17.1.99
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 neighbor 192.168.3.2 description finance
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
 !
 address-family ipv4 multicast
  neighbor 192.168.3.2 activate
  neighbor 192.168.3.2 advertisement-interval 25
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

Examples: Removing BGP Configuration Commands Using a Redistribution Example

The following examples show first the CLI configuration to enable the redistribution of BGP routes into EIGRP using a route map and then the CLI configuration to remove the redistribution and route map. Some BGP configuration commands can affect other CLI commands and this example demonstrates how the removal of one command affects another command.

In the first configuration example, a route map is configured to match and set autonomous system numbers. BGP neighbors in three different autonomous systems are configured and activated. An EIGRP routing process is started, and the redistribution of BGP routes into EIGRP using the route map is configured.

CLI to Enable BGP Route Redistribution Into EIGRP

```
route-map bgp-to-eigrp permit 10
 match tag 50000
```

```
 set tag 65000
 exit
router bgp 45000
 bgp log-neighbor-changes
 address-family ipv4
  neighbor 172.16.1.2 remote-as 45000
  neighbor 172.21.1.2 remote-as 45000
  neighbor 192.168.1.2 remote-as 40000
  neighbor 192.168.3.2 remote-as 50000
  neighbor 172.16.1.2 activate
  neighbor 172.21.1.2 activate
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
 exit
router eigrp 100
 redistribute bgp 45000 metric 10000 100 255 1 1500 route-map bgp-to-eigrp
 no auto-summary
 exit
```

In the second configuration example, both the **route-map** command and the **redistribute** command are disabled. If only the route-map command is removed, it does not automatically disable the redistribution. The redistribution will now occur without any matching or filtering. To remove the redistribution configuration, the **redistribute** command must also be disabled.

CLI to Remove BGP Route Redistribution Into EIGRP

```
configure terminal
 no route-map bgp-to-eigrp
 router eigrp 100
  no redistribute bgp 45000
  end
```

Examples: BGP Soft Reset

The following examples show two ways to reset the connection for BGP peer 192.168.1.1.

Example: Dynamic Inbound Soft Reset

The following example shows the command used to initiate a dynamic soft reconfiguration in the BGP peer 192.168.1.1. This command requires that the peer support the route refresh capability.

```
clear ip bgp 192.168.1.1 soft in
```

Example: Inbound Soft Reset Using Stored Information

The following example shows how to enable inbound soft reconfiguration for the neighbor 192.168.1.1. All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is performed later, the stored information will be used to generate a new set of inbound updates.

```
router bgp 100
 neighbor 192.168.1.1 remote-as 200
 neighbor 192.168.1.1 soft-reconfiguration inbound
```

The following example clears the session with the neighbor 192.168.1.1:

```
clear ip bgp 192.168.1.1 soft in
```

Example: Resetting and Displaying Basic BGP Information

The following example shows how to reset and display basic BGP information.

The **clear ip bgp *** command clears and resets all the BGP neighbor sessions. Specific neighbors or all peers in an autonomous system can be cleared by using the *neighbor-address* and *autonomous-system-number* arguments. If no argument is specified, this command will clear and reset all BGP neighbor sessions.

Note The **clear ip bgp *** command also clears all the internal BGP structures, which makes it useful as a troubleshooting tool.

```
Device# clear ip bgp *
```

The **show ip bgp** command is used to display all the entries in the BGP routing table. The following example displays BGP routing table information for the 10.1.1.0 network:

```
Device# show ip bgp 10.1.1.0 255.255.255.0

BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  40000
    192.168.1.2 from 192.168.1.2 (10.1.1.99)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

The **show ip bgp neighbors** command is used to display information about the TCP and BGP connections to neighbors. The following example displays the routes that were advertised from Router B in the figure above (in the “Configuring a BGP Peer for the IPv4 VRF Address Family” section) to its BGP neighbor 192.168.3.2 on Router E:

```
Device# show ip bgp neighbors 192.168.3.2 advertised-routes

BGP table version is 3, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 40000 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
Total number of prefixes 2
```

The **show ip bgp paths** command is used to display all the BGP paths in the database. The following example displays BGP path information for Router B in the figure above (in the “Customizing a BGP Peer” section):

```
Device# show ip bgp paths

Address    Hash Refcount Metric Path
0x2FB5DB0     0        5      0 i
0x2FB5C90     1        4      0 i
0x2FB5C00  1361        2      0 50000 i
0x2FB5D20  2625        2      0 40000 i
```

The **show ip bgp summary** command is used to display the status of all BGP connections. The following example displays BGP routing table information for Router B in the figure above (in the "Customizing a BGP Peer" section):

```
Device# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 45000
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
4/2 BGP path/bestpath attribute entries using 496 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 882 total bytes of memory
BGP activity 14/10 prefixes, 16/12 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.1.2     4 40000     667     672        3    0    0 00:03:49        1
192.168.3.2     4 50000     468     467        0    0    0 00:03:49 (NoNeg)
```

Examples: Aggregating Prefixes Using BGP

The following examples show how you can use aggregate routes in BGP either by redistributing an aggregate route into BGP or by using the BGP conditional aggregation routing feature.

In the following example, the **redistribute static** router configuration command is used to redistribute aggregate route 10.0.0.0:

```
ip route 10.0.0.0 255.0.0.0 null 0
!
router bgp 100
 redistribute static
```

The following configuration shows how to create an aggregate entry in the BGP routing table when at least one specific route falls into the specified range. The aggregate route will be advertised as coming from your autonomous system and has the atomic aggregate attribute set to show that information might be missing. (By default, atomic aggregate is set unless you use the **as-set** keyword in the **aggregate-address** router configuration command.)

```
router bgp 100
 aggregate-address 10.0.0.0 255.0.0.0
```

The following example shows how to create an aggregate entry using the same rules as in the previous example, but the path advertised for this route will be an AS_SET consisting of all elements contained in all paths that are being summarized:

```
router bgp 100
 aggregate-address 10.0.0.0 255.0.0.0 as-set
```

The following example shows how to create the aggregate route for 10.0.0.0 and also suppress advertisements of more specific routes to all neighbors:

```
router bgp 100
 aggregate-address 10.0.0.0 255.0.0.0 summary-only
```

The following example configures BGP to not advertise inactive routes:

```
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 unicast
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```

The following example configures a maximum route limit in the VRF named RED and configures BGP to not advertise inactive routes through the VRF named RED:

```
Device(config)# ip vrf RED
Device(config-vrf)# rd 50000:10
Device(config-vrf)# maximum routes 1000 10
Device(config-vrf)# exit
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 vrf RED
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```

Example: Configuring a BGP Peer Group

The following example shows how to use an address family to configure a peer group so that all members of the peer group are both unicast- and multicast-capable:

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 address-family ipv4 unicast
  neighbor mygroup peer-group
  neighbor 192.168.1.2 peer-group mygroup
  neighbor 192.168.3.2 peer-group mygroup
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 address-family ipv4 multicast
  neighbor mygroup peer-group
  neighbor 192.168.1.2 peer-group mygroup
  neighbor 192.168.3.2 peer-group mygroup
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration on an Autonomous System (AS)*
RFC 2519	*A Framework for Inter-Domain Route Aggregation*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 4271	*A Border Gateway Protocol 4 (BGP-4)*

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

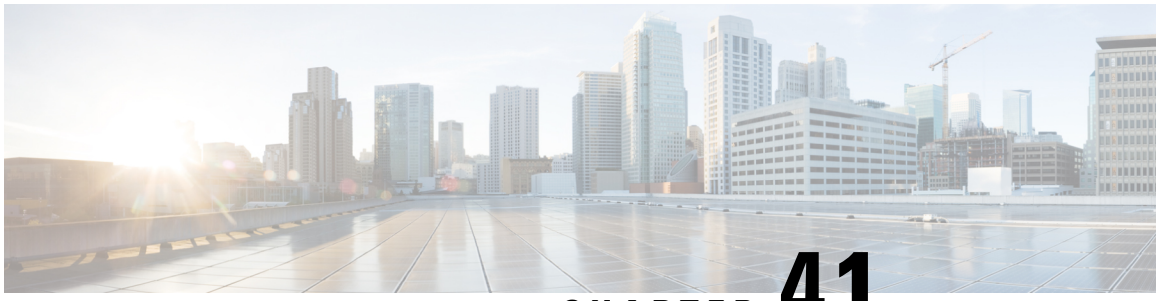
Feature Information for BGP 4

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 54: Feature Information for BGP 4

Feature Name	Releases	Feature Information
BGP 4	Cisco IOS XE Release 2.1	BGP is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco software implementation of BGP Version 4 includes multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families, including IP Version 4 (IPv4), IP Version 6 (IPv6), Virtual Private Networks version 4 (VPNv4), and Connectionless Network Services (CLNS).



CHAPTER 41

Configuring a Basic BGP Network

This module describes the basic tasks to configure a basic Border Gateway Protocol (BGP) network. BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. The Cisco IOS implementation of the neighbor and address family commands is explained. This module also contains tasks to configure and customize BGP peers, implement BGP route aggregation, configure BGP route origination, and define BGP backdoor routes. BGP peer group definition is documented, peer session templates are introduced, and update groups are explained,

Prerequisites for Configuring a Basic BGP Network

Before configuring a basic BGP network, you should be familiar with the “Cisco BGP Overview” module.

Restrictions for Configuring a Basic BGP Network

A device that runs Cisco software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.

Information About Configuring a Basic BGP Network

BGP Version 4

Border Gateway Protocol (BGP) is an interdomain routing protocol designed to provide loop-free routing between separate routing domains that contain independent routing policies (autonomous systems). The Cisco

software implementation of BGP version 4 includes multiprotocol extensions to allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families including IP Version 4 (IPv4), IP Version 6 (IPv6), and Virtual Private Networks version 4 (VPNv4).

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. When connecting to an external organization, external BGP (eBGP) peering sessions are created. Although BGP is referred to as an exterior gateway protocol (EGP) many networks within an organization are becoming so complex that BGP can be used to simplify the internal network used within the organization. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

Note BGP requires more configuration than other routing protocols, and the effects of any configuration changes must be fully understood. Incorrect configuration can create routing loops and negatively impact normal network operation.

BGP Router ID

BGP uses a router ID to identify BGP-speaking peers. The BGP router ID is a 32-bit value that is often represented by an IPv4 address. By default, the Cisco software sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the device, the software chooses the highest IPv4 address configured on a physical interface of the device to represent the BGP router ID. The BGP router ID must be unique to the BGP peers in a network.

BGP-Speaker and Peer Relationships

A BGP-speaking device does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking devices. A peer device is a BGP-speaking device that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a neighbor, but because this can imply the idea that the BGP devices are directly connected with no other device in between, the term *neighbor* will be avoided whenever possible in this document. A BGP speaker is the local device, and a peer is any other BGP-speaking network device.

When a TCP connection is established between peers, each BGP peer initially exchanges all its routes—the complete BGP routing table—with the other peer. After this initial exchange, only incremental updates are sent when there has been a topology change in the network, or when a routing policy has been implemented or modified. In the periods of inactivity between these updates, peers exchange special messages called keepalives.

A BGP autonomous system is a network that is controlled by a single technical administration entity. Peer devices are called external peers when they are in different autonomous systems and internal peers when they are in the same autonomous system. Usually, external peers are adjacent and share a subnet; internal peers may be anywhere in the same autonomous system.

BGP Autonomous System Number Formats

Prior to January 2009, BGP autonomous system numbers that were allocated to companies were 2-octet numbers in the range from 1 to 65535 as described in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)* . Due to increased demand for autonomous system numbers, the Internet Assigned Number Authority (IANA) will start in January 2009 to allocate four-octet autonomous system numbers in the range from 65536 to

4294967295. RFC 5396, *Textual Representation of Autonomous System (AS) Numbers* , documents three methods of representing autonomous system numbers. Cisco has implemented the following two methods:

- Asplain--Decimal value notation where both 2-byte and 4-byte autonomous system numbers are represented by their decimal value. For example, 65526 is a 2-byte autonomous system number and 234567 is a 4-byte autonomous system number.
- Asdot--Autonomous system dot notation where 2-byte autonomous system numbers are represented by their decimal value and 4-byte autonomous system numbers are represented by a dot notation. For example, 65526 is a 2-byte autonomous system number and 1.169031 is a 4-byte autonomous system number (this is dot notation for the 234567 decimal number).

For details about the third method of representing autonomous system numbers, see RFC 5396.

Asdot Only Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases, the 4-octet (4-byte) autonomous system numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte autonomous system numbers the asdot format includes a period which is a special character in regular expressions. A backslash must be entered before the period (for example, 1\.14) to ensure the regular expression match does not fail. The table below shows the format in which 2-byte and 4-byte autonomous system numbers are configured, matched in regular expressions, and displayed in **show** command output in Cisco IOS images where only asdot formatting is available.

Table 55: Asdot Only 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte autonomous system numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte autonomous system numbers are written in the asplain format. If you want to change the default **show** command output to display 4-byte autonomous system numbers in the asdot format, use the **bgp asnotation dot** command under router configuration mode. When the asdot format is enabled as the default, any regular expressions to match 4-byte autonomous system numbers must be written using the asdot format, or the regular expression match will fail. The tables below show that although you can configure 4-byte autonomous system numbers in either asplain or asdot format, only one format is used to display **show** command output and control 4-byte autonomous system number matching for regular expressions, and the default is asplain format. To display 4-byte autonomous system numbers in **show** command output and to control matching for regular expressions in the asdot format, you must configure the **bgp asnotation dot** command. After enabling the **bgp asnotation dot** command, a hard reset must be initiated for all BGP sessions by entering the **clear ip bgp *** command.

Note If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The **show** command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

Table 56: Default Asplain 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 57: Asdot 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private Autonomous System Numbers

In Cisco IOS Release 12.0(32)S12, 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 12.4(24)T, and later releases, the Cisco implementation of BGP supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use* , describes new reserved autonomous system numbers for documentation purposes. Use of the reserved numbers allow configuration examples to be accurately documented and avoids conflict with production networks if these configurations are literally copied. The reserved numbers are documented in the IANA autonomous system number registry. Reserved 2-byte autonomous system numbers are in the contiguous block, 64496 to 64511 and reserved 4-byte autonomous system numbers are from 65536 to 65551 inclusive.

Private 2-byte autonomous system numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private autonomous system numbers can be used for internal routing domains but must be translated for traffic that is routed out to the Internet. BGP should not be configured to advertise private autonomous system numbers to external networks. Cisco IOS software does not remove private autonomous system numbers from routing updates by default. We recommend that ISPs filter private autonomous system numbers.

Note Autonomous system number assignment for public and private networks is governed by the IANA. For information about autonomous-system numbers, including reserved number assignment, or to apply to register an autonomous system number, see the following URL: http://www.iana.org/.

Cisco Implementation of 4-Byte Autonomous System Numbers

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 15.1(1)SG, and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain (65538, for example) as the default regular expression match and the output display format for AS numbers. However, you can configure 4-byte autonomous system numbers in both the asplain format and the asdot format as described in RFC 5396.

To change the default regular expression match and output display of 4-byte autonomous system numbers to asdot format, use the **bgp asnotation dot** command followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions.

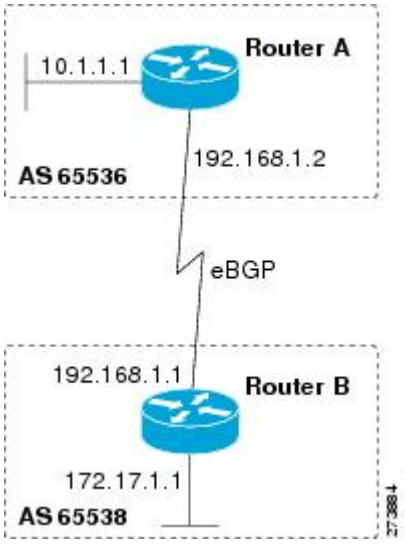
In Cisco IOS Release 12.0(32)S12, and 12.4(24)T, the Cisco implementation of 4-byte autonomous system numbers uses asdot (1.2, for example) as the only configuration format, regular expression match, and output display, with no asplain support.

For an example of BGP peers in two autonomous systems using 4-byte numbers, see the figure below. To view a configuration example of the configuration between three neighbor peers in separate 4-byte autonomous systems configured using asdot notation, see the Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers.

Cisco also supports RFC 4893, which was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. To ensure a smooth transition, we recommend that all BGP speakers within an autonomous system that is identified using a 4-byte autonomous system number be upgraded to support 4-byte autonomous system numbers.

Note A new private autonomous system number, 23456, was created by RFC 4893, and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

Figure 40: BGP Peers in Two Autonomous Systems Using 4-Byte Numbers



BGP Peer Session Establishment

When a BGP routing process establishes a peering session with a peer, it goes through the following state changes:

- Idle—The initial state that the BGP routing process enters when the routing process is enabled or when the device is reset. In this state, the device waits for a start event, such as a peering configuration with a remote peer. After the device receives a TCP connection request from a remote peer, the device initiates another start event to wait for a timer before starting a TCP connection to a remote peer. If the device is reset, the peer is reset and the BGP routing process returns to the Idle state.
- Connect—The BGP routing process detects that a peer is trying to establish a TCP session with the local BGP speaker.
- Active—In this state, the BGP routing process tries to establish a TCP session with a peer device using the ConnectRetry timer. Start events are ignored while the BGP routing process is in the Active state. If the BGP routing process is reconfigured or if an error occurs, the BGP routing process will release system resources and return to an Idle state.
- OpenSent—The TCP connection is established, and the BGP routing process sends an OPEN message to the remote peer, and transitions to the OpenSent state. The BGP routing process can receive other OPEN messages in this state. If the connection fails, the BGP routing process transitions to the Active state.
- OpenReceive—The BGP routing process receives the OPEN message from the remote peer and waits for an initial keepalive message from the remote peer. When a keepalive message is received, the BGP routing process transitions to the Established state. If a notification message is received, the BGP routing process transitions to the Idle state. If an error or configuration change occurs that affects the peering session, the BGP routing process sends a notification message with the Finite State Machine (FSM) error code and then transitions to the Idle state.
- Established—The initial keepalive is received from the remote peer. Peering is now established with the remote neighbor and the BGP routing process starts exchanging update message with the remote peer.

The hold timer restarts when an update or keepalive message is received. If the BGP process receives an error notification, it will transition to the Idle state.

Cisco Implementation of BGP Global and Address Family Configuration Commands

The address family model for configuring BGP is based on splitting apart the configuration for each address family. All commands that are independent of the address family are grouped together at the beginning (highest level) of the configuration, and these are followed by separate submodes for commands specific to each address family (with the exception that commands relating to IPv4 unicast can also be entered at the beginning of the configuration). When a network operator configures BGP, the flow of BGP configuration categories is represented by the following bullets in order:

- Global configuration—Configuration that is applied to BGP in general, rather than to specific neighbors. For example, the **network**, **redistribute**, and **bgp bestpath** commands.
- Address family-dependent configuration—Configuration that applies to a specific address family such as policy on an individual neighbor.

The relationship between BGP global and BGP address family-dependent configuration categories is shown in the table below.

Table 58: Relationships Between BGP Configuration Categories

BGP Configuration Category	Configuration Sets Within Category
Global address family-independent	One set of global address family-independent configurations
Address family-dependent	One set of global address family-dependent configurations per address family

Note Address family configuration must be entered within the address family submode to which it applies.

The following is an example of BGP configuration statements showing the grouping of global address family-independent and address family-dependent commands.

```
router bgp <AS>
 ! AF independent part
 neighbor <ip-address> <command> ! Session config; AF independent
 address-family ipv4 unicast
  ! AF dependant part
  neighbor <ip-address> <command> ! Policy config; AF dependant
  exit-address-family
 address-family ipv4 multicast
  ! AF dependant part
  neighbor <ip-address> <command> ! Policy config; AF dependant
  exit-address-family
 address-family ipv4 unicast vrf <vrf-name>
  ! VRF specific AS independent commands
  ! VRF specific AS dependant commands
  neighbor <ip-address> <command> ! Session config; AF independent
```

```
  neighbor <ip-address> <command> ! Policy config; AF dependant
  exit-address-family
```

The following example shows actual BGP commands that match the BGP configuration statements in the previous example:

```
router bgp 45000
 router-id 172.17.1.99
 bgp log-neighbor-changes
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 address-family ipv4 unicast
  neighbor 192.168.1.2 activate
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
 address-family ipv4 multicast
  neighbor 192.168.3.2 activate
  neighbor 192.168.3.2 advertisement-interval 25
  network 172.16.1.0 mask 255.255.255.0
  exit-address-family
 address-family ipv4 vrf vpn1
  neighbor 192.168.3.2 activate
  network 172.21.1.0 mask 255.255.255.0
  exit-address-family
```

The **bgp upgrade-cli** command simplifies the migration of BGP networks and existing configurations from the network layer reachability information (NLRI) format to the address family format. Network operators can configure commands in the address family identifier (AFI) format and save these command configurations to existing NLRI formatted configurations. The BGP hybrid command-line interface (CLI) does not add support for complete AFI and NLRI integration because of the limitations of the NLRI format. For complete support of AFI commands and features, we recommend upgrading existing NLRI configurations with the **bgp upgrade-cli** command. For an example of migrating BGP configurations from the NLRI format to the address family format, see the “Example: NLFI to AFI Configuration” section later in this module.

BGP Session Reset

Whenever the routing policy changes due to a configuration change, BGP peering sessions must be reset by using the **clear ip bgp** command. Cisco software supports the following three mechanisms to reset BGP peering sessions:

- Hard reset—A hard reset tears down the specified peering sessions including the TCP connection and deletes routes coming from the specified peer.
- Soft reset—A soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions. Soft reconfiguration uses stored update information, at the cost of additional memory for storing the updates, to allow you to apply new BGP policy without disrupting the network. Soft reconfiguration can be configured for inbound or outbound sessions.
- Dynamic inbound soft reset—The route refresh capability, as defined in RFC 2918, allows the local device to reset inbound routing tables dynamically by exchanging route refresh requests to supporting peers. The route refresh capability does not store update information locally for nondisruptive policy changes. It instead relies on dynamic exchange with supporting peers. Route refresh must first be advertised through BGP capability negotiation between peers. All BGP devices must support the route refresh capability. To determine if a BGP device supports this capability, use the **show ip bgp neighbors** command. The following message is displayed in the output when the device supports the route refresh capability:

```
Received route refresh capability from peer.
```

The **bgp soft-reconfig-backup** command was introduced to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command.

BGP Route Aggregation

BGP peers store and exchange routing information and the amount of routing information increases as more BGP speakers are configured. The use of route aggregation reduces the amount of information involved. Aggregation is the process of combining the attributes of several different routes so that only a single route is advertised. Aggregate prefixes use the classless interdomain routing (CIDR) principle to combine contiguous networks into one classless set of IP addresses that can be summarized in routing tables. Fewer routes now need to be advertised.

Two methods are available in BGP to implement route aggregation. You can redistribute an aggregated route into BGP or you can use a form of conditional aggregation. Basic route redistribution involves creating an aggregate route and then redistributing the routes into BGP. Conditional aggregation involves creating an aggregate route and then advertising or suppressing the advertising of certain routes on the basis of route maps, autonomous system set path (AS-SET) information, or summary information.

The **bgp suppress-inactive** command configures BGP to not advertise inactive routes to any BGP peer. A BGP routing process can advertise routes that are not installed in the routing information database (RIB) to BGP peers by default. A route that is not installed into the RIB is an inactive route. Inactive route advertisement can occur, for example, when routes are advertised through common route aggregation. Inactive route advertisements can be suppressed to provide more consistent data forwarding.

BGP Aggregation Route AS_SET Information Generation

AS_SET information can be generated when BGP routes are aggregated using the **aggregate-address** command. The path advertised for such a route is an AS_SET consisting of all the elements, including the communities, contained in all the paths that are being summarized. If the AS_PATHs to be aggregated are identical, only the AS_PATH is advertised. The ATOMIC_AGGREGATE attribute, set by default for the **aggregate-address** command, is not added to the AS_SET.

Routing Policy Change Management

Routing policies for a peer include all the configurations for elements such as route map, distribute list, prefix list, and filter list that may impact inbound or outbound routing table updates. The policy changes are automatically updated to peers whenever there is a change in the routing policy. Performing inbound reset enables the new inbound policy configured on the router to take effect. Performing outbound reset causes the new local outbound policy configured on the router to take effect without resetting the BGP session. As a new set of updates is sent during outbound policy reset, a new inbound policy of the neighbor can also take effect. This means that after changing inbound policy you must do an inbound reset on the local router or an outbound reset on the peer router. Outbound policy changes require an outbound reset on the local router or an inbound reset on the peer router.

There are two types of reset: hard reset and soft reset. The table below lists their advantages and disadvantages.

Table 59: Advantages and Disadvantages of Hard and Soft Resets

Type of Reset	Advantages	Disadvantages
Hard reset	No memory overhead.	The prefixes in the BGP, IP, and Forwarding Information Base (FIB) tables provided by the neighbor are lost. Not recommended.
Outbound soft reset	No configuration, no storing of routing table updates.	Does not reset inbound routing table updates.
Dynamic inbound soft reset	Does not clear the BGP session and cache. Does not require storing of routing table updates, and has no memory overhead.	Both BGP routers must support the route refresh capability (in Cisco IOS Release 12.1 and later releases). **Note** Does not reset outbound routing table updates.
Configured inbound soft reset (uses the **neighbor soft-reconfiguration** router configuration command)	Can be used when both BGP routers do not support the automatic route refresh capability. In Cisco IOS Release 12.3(14)T, the **bgp soft-reconfig-backup** command was introduced to configure inbound soft reconfiguration for peers that do not support the route refresh capability.	Requires preconfiguration. Stores all received (inbound) routing policy updates without modification; is memory-intensive. Recommended only when absolutely necessary, such as when both BGP routers do not support the automatic route refresh capability. **Note** Does not reset outbound routing table updates.

Once you have defined two routers to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you must reset BGP connections for the configuration change to take effect.

A soft reset updates the routing table for inbound and outbound routing updates. Cisco IOS Release 12.1 and later releases support soft reset without any prior configuration. This soft reset allows the dynamic exchange of route refresh requests and routing information between BGP routers, and the subsequent readvertisement of the respective outbound routing table. There are two types of soft reset:

- When soft reset is used to generate inbound updates from a neighbor, it is called dynamic inbound soft reset.
- When soft reset is used to send a new set of updates to a neighbor, it is called outbound soft reset.

To use soft reset without preconfiguration, both BGP peers must support the soft route refresh capability, which is advertised in the OPEN message sent when the peers establish a TCP session. Routers running Cisco IOS releases prior to Release 12.1 do not support the route refresh capability and must clear the BGP session using the **neighbor soft-reconfiguration** router configuration command. Clearing the BGP session in this way will have a negative impact upon network operations and should be used only as a last resort.

Conditional BGP Route Injection

Routes that are advertised through the BGP are commonly aggregated to minimize the number of routes that are used and reduce the size of global routing tables. However, common route aggregation can obscure more specific routing information that is more accurate but not necessary to forward packets to their destinations. Routing accuracy is obscured by common route aggregation because a prefix that represents multiple addresses or hosts over a large topological area cannot be accurately reflected in a single route. Cisco software provides several methods by which you can originate a prefix into BGP. Prior to the BGP conditional route injection feature, the existing methods included redistribution and using the **network** or **aggregate-address** command. However, these methods assume the existence of more specific routing information (matching the route to be originated) in either the routing table or the BGP table.

BGP conditional route injection allows you to originate a prefix into a BGP routing table without the corresponding match. This feature allows more specific routes to be generated based on administrative policy or traffic engineering information in order to provide more specific control over the forwarding of packets to these more specific routes, which are injected into the BGP routing table only if the configured conditions are met. Enabling this feature will allow you to improve the accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes. Only prefixes that are equal to or more specific than the original prefix may be injected. BGP conditional route injection is enabled with the **bgp inject-map exist-map**command and uses two route maps (inject map and exist map) to install one (or more) more specific prefixes into a BGP routing table. The exist map specifies the prefixes that the BGP speaker will track. The inject map defines the prefixes that will be created and installed into the local BGP table.

Note Inject maps and exist maps will only match a single prefix per route map clause. To inject additional prefixes, you must configure additional route map clauses. If multiple prefixes are used, the first prefix matched will be used.

BGP Peer Groups

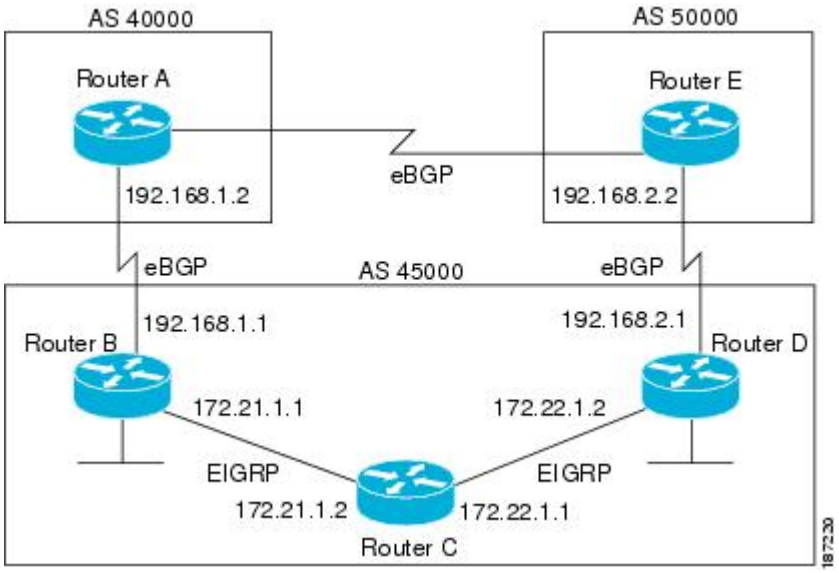
Often, in a BGP network, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into BGP peer groups to simplify configuration and, more importantly, to make configuration updates more efficient. When you have many peers, this approach is highly recommended.

BGP Backdoor Routes

In a BGP network topology with two border devices using eBGP to communicate to a number of different autonomous systems, using eBGP to communicate between the two border devices may not be the most efficient routing method. In the figure below, Router B as a BGP speaker will receive a route to Router D through eBGP, but this route will traverse at least two autonomous systems. Router B and Router D are also connected through an Enhanced Interior Gateway Routing Protocol (EIGRP) network (any IGP can be used here), and this route has a shorter path. EIGRP routes, however, have a default administrative distance of 90, and eBGP routes have a default administrative distance of 20, so BGP will prefer the eBGP route. Changing the default administrative distances is not recommended because changing the administrative distance may lead to routing loops. To cause BGP to prefer the EIGRP route, you can use the **network backdoor** command. BGP treats the network specified by the **network backdoor** command as a locally assigned network, except

that it does not advertise the specified network in BGP updates. In the figure below, this means that Router B will communicate to Router D using the shorter EIGRP route instead of the longer eBGP route.

Figure 41: BGP Backdoor Route Topology



Peer Groups and BGP Update Messages

In Cisco IOS software releases prior to Release 12.0(24)S, 12.2(18)S, or 12.3(4)T, BGP update messages were grouped based on peer group configurations. This method of grouping neighbors for BGP update message generation reduced the amount of system processing resources needed to scan the routing table. This method, however, had the following limitations:

- All neighbors that shared peer group configuration also had to share outbound routing policies.
- All neighbors had to belong to the same peer group and address family. Neighbors configured in different address families could not belong to different peer groups.

These limitations existed to balance optimal update generation and replication against peer group configuration. These limitations could cause the network operator to configure smaller peer groups, which reduced the efficiency of update message generation and limited the scalability of neighbor configuration.

BGP Update Group

The introduction of the BGP (dynamic) update group provides a different type of BGP peer grouping from existing BGP peer groups. Existing peer groups are not affected but peers with the same outbound policy configured that are not members of a current peer group can be grouped into an update group. The members of this update group will use the same update generation engine. When BGP update groups are configured an algorithm dynamically calculates the BGP update group membership based on outbound policies. Optimal BGP update message generation occurs automatically and independently. BGP neighbor configuration is no longer restricted by outbound routing policies, and update groups can belong to different address families.

BGP Dynamic Update Group Configuration

In Cisco IOS Release 12.0(24)S, 12.2(18)S, 12.3(4)T, 12.2(27)SBC, and later releases, a new algorithm was introduced that dynamically calculates and optimizes update groups of neighbors that share the same outbound policies and can share the same update messages. No configuration is required to enable the BGP dynamic update group and the algorithm runs automatically. When a change to outbound policy occurs, the router automatically recalculates update group memberships and applies the changes by triggering an outbound soft reset after a 1-minute timer expires. This behavior is designed to provide the network operator with time to change the configuration if a mistake is made. You can manually enable an outbound soft reset before the timer expires by entering the **clear ip bgp** *ip-address* **soft out**command.

Note In Cisco IOS Release 12.0(22)S, 12.2(14)S, 12.3(2)T, and prior releases, the update group recalculation delay timer is set to 3 minutes.

For the best optimization of BGP update group generation, we recommend that the network operator keeps outbound routing policy the same for neighbors that have similar outbound policies.

BGP Peer Templates

To address some of the limitations of peer groups such as configuration management, BGP peer templates were introduced to support the BGP update group configuration.

A peer template is a configuration pattern that can be applied to neighbors that share policies. Peer templates are reusable and support inheritance, which allows the network operator to group and apply distinct neighbor configurations for BGP neighbors that share policies. Peer templates also allow the network operator to define very complex configuration patterns through the capability of a peer template to inherit a configuration from another peer template.

There are two types of peer templates:

- Peer session templates are used to group and apply the configuration of general session commands that are common to all address family and NLRI configuration modes.
- Peer policy templates are used to group and apply the configuration of commands that are applied within specific address families and NLRI configuration modes.

Peer templates improve the flexibility and enhance the capability of neighbor configuration. Peer templates also provide an alternative to peer group configuration and overcome some limitations of peer groups. BGP peer routers using peer templates also benefit from automatic update group configuration. With the configuration of the BGP peer templates and the support of the BGP dynamic update peer groups, the network operator no longer needs to configure peer groups in BGP and the network can benefit from improved configuration flexibility and faster convergence.

Note A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies from peer templates.

The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.

- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

Inheritance in Peer Templates

The inheritance capability is a key component of peer template operation. Inheritance in a peer template is similar to node and tree structures commonly found in general computing, for example, file and directory trees. A peer template can directly or indirectly inherit the configuration from another peer template. The directly inherited peer template represents the tree in the structure. The indirectly inherited peer template represents a node in the tree. Because each node also supports inheritance, branches can be created that apply the configurations of all indirectly inherited peer templates within a chain back to the directly inherited peer template or the source of the tree.

This structure eliminates the need to repeat configuration statements that are commonly reapplied to groups of neighbors because common configuration statements can be applied once and then indirectly inherited by peer templates that are applied to neighbor groups with common configurations. Configuration statements that are duplicated separately within a node and a tree are filtered out at the source of the tree by the directly inherited template. A directly inherited template will overwrite any indirectly inherited statements that are duplicated in the directly inherited template.

Inheritance expands the scalability and flexibility of neighbor configuration by allowing you to chain together peer templates configurations to create simple configurations that inherit common configuration statements or complex configurations that apply very specific configuration statements along with common inherited configurations. Specific details about configuring inheritance in peer session templates and peer policy templates are provided in the following sections.

When BGP neighbors use inherited peer templates it can be difficult to determine which policies are associated with a specific template. The **detail** keyword was added to the **show ip bgp template peer-policy** command to display the detailed configuration of local and inherited policies associated with a specific template.

Peer Session Templates

Peer session templates are used to group and apply the configuration of general session commands to groups of neighbors that share session configuration elements. General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode. Only general session commands can be configured in a peer session template. The following general session commands are supported by peer session templates:

- **description**
- **disable-connected-check**
- **ebgp-multihop**
- **exit peer-session**
- **inherit peer-session**
- **local-as**
- **password**
- **remote-as**

- **shutdown**
- **timers**
- **translate-update**
- **update-source**
- **version**

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Peer session templates support direct and indirect inheritance. A peer can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template.

Note If you attempt to configure more than one inherit statement with a single peer session template, an error message will be displayed.

This behavior allows a BGP neighbor to directly inherit only one session template and indirectly inherit up to seven additional peer session templates. This allows you to apply up to a maximum of eight peer session configurations to a neighbor: the configuration from the directly inherited peer session template and the configurations from up to seven indirectly inherited peer session templates. Inherited peer session configurations are evaluated first and applied starting with the last node in the branch and ending with the directly applied peer session template configuration at the source of the tree. The directly applied peer session template will have priority over inherited peer session template configurations. Any configuration statements that are duplicated in inherited peer session templates will be overwritten by the directly applied peer session template. So, if a general session command is reapplied with a different value, the subsequent value will have priority and overwrite the previous value that was configured in the indirectly inherited template. The following examples illustrate the use of this feature.

In the following example, the general session command **remote-as 1** is applied in the peer session template named SESSION-TEMPLATE-ONE:

```
template peer-session SESSION-TEMPLATE-ONE
 remote-as 1
 exit peer-session
```

Peer session templates support only general session commands. BGP policy configuration commands that are configured only for a specific address family or NLRI configuration mode are configured with peer policy templates.

BGP Peer Templates

To address some of the limitations of peer groups such as configuration management, BGP peer templates were introduced to support the BGP update group configuration.

A peer template is a configuration pattern that can be applied to neighbors that share policies. Peer templates are reusable and support inheritance, which allows the network operator to group and apply distinct neighbor configurations for BGP neighbors that share policies. Peer templates also allow the network operator to define

very complex configuration patterns through the capability of a peer template to inherit a configuration from another peer template.

There are two types of peer templates:

- Peer session templates are used to group and apply the configuration of general session commands that are common to all address family and NLRI configuration modes.
- Peer policy templates are used to group and apply the configuration of commands that are applied within specific address families and NLRI configuration modes.

Peer templates improve the flexibility and enhance the capability of neighbor configuration. Peer templates also provide an alternative to peer group configuration and overcome some limitations of peer groups. BGP peer routers using peer templates also benefit from automatic update group configuration. With the configuration of the BGP peer templates and the support of the BGP dynamic update peer groups, the network operator no longer needs to configure peer groups in BGP and the network can benefit from improved configuration flexibility and faster convergence.

Note A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies from peer templates.

The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

BGP IPv6 Neighbor Activation Under the IPv4 Address Family

Prior to Cisco IOS Release 12.2(33)SRE4, by default, both IPv6 and IPv4 capability is exchanged with a BGP peer that has an IPv6 address. When an IPv6 peer is configured, that neighbor is automatically activated under the IPv4 unicast address family.

Beginning with Cisco IOS Release 12.2(33)SRE4, when a *new* IPv6 neighbor is being configured, it is no longer automatically activated under the IPv4 address family. You can manually activate the IPv6 neighbor under the IPv4 address family if, for example, you have a dual stack environment and want to send IPv6 and IPv4 prefixes.

If you do not want an *existing* IPv6 peer to be activated under the IPv4 address family, you can manually deactivate the peer with the **no neighbor activate** command. Until then, existing configurations that activate an IPv6 neighbor under the IPv4 unicast address family will continue to try to establish a session.

How to Configure a Basic BGP Network

Configuring a basic BGP network consists of a few required tasks and many optional tasks. A BGP routing process must be configured and BGP peers must be configured, preferably using the address family configuration model. If the BGP peers are part of a VPN network, the BGP peers must be configured using the IPv4 VRF address family task. The other tasks in the following list are optional:

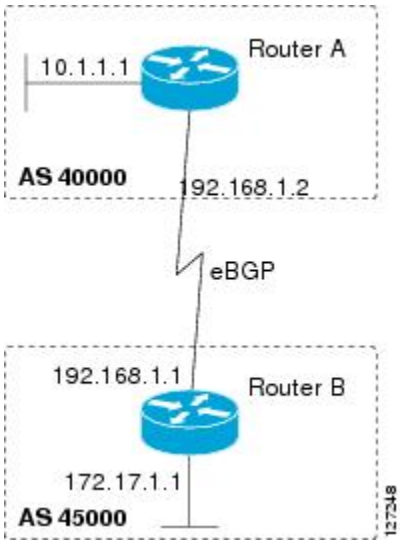
Configuring a BGP Routing Process

Perform this task to configure a BGP routing process. You must perform the required steps at least once to enable BGP. The optional steps here allow you to configure additional features in your BGP network. Several of the features, such as logging neighbor resets and immediate reset of a peer when its link goes down, are enabled by default but are presented here to enhance your understanding of how your BGP network operates.

Note A device that runs Cisco software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple concurrent BGP address family and subaddress family configurations.

The configuration in this task is done at Router A in the figure below and would need to be repeated with appropriate changes to the IP addresses (for example, at Router B) to fully achieve a BGP process between the two devices. No address family is configured here for the BGP routing process, so routing information for the IPv4 unicast address family is advertised by default.

Figure 42: BGP Topology with Two Autonomous Systems



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
5. **bgp router-id** *ip-address*
6. **timers bgp** *keepalive holdtime*
7. **bgp fast-external-fallover**
8. **bgp log-neighbor-changes**
9. **end**
10. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Configures a BGP routing process, and enters router configuration mode for the specified routing process. • Use the *autonomous-system-number* argument to specify an integer, from 0 and 65534, that identifies the device to other BGP speakers.
Step 4	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router)# network 10.1.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols, the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 5	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id 10.1.1.99`	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. • Use the *ip-address* argument to specify a unique router ID within the network. **Note** Configuring a router ID using the **bgp router-id** command resets all active BGP peering sessions.
Step 6	**timers bgp** *keepalive holdtime* **Example:** `Device(config-router)# timers bgp 70 120`	(Optional) Sets BGP network timers. • Use the *keepalive* argument to specify the frequency, in seconds, with which the software sends keepalive messages to its BGP peer. By default, the keepalive timer is set to 60 seconds. • Use the *holdtime* argument to specify the interval, in seconds, after which the software, having not received a keepalive message, declares a BGP peer dead. By default, the holdtime timer is set to 180 seconds.
Step 7	**bgp fast-external-fallover** **Example:**	(Optional) Enables the automatic resetting of BGP sessions.

	Command or Action	Purpose
	`Device(config-router)# bgp fast-external-fallover`	• By default, the BGP sessions of any directly adjacent external peers are reset if the link used to reach them goes down.
Step 8	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets. • Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 10	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output from the **show ip bgp** command shows the BGP routing table for Router A in the figure above after this task has been configured on Router A. You can see an entry for the network 10.1.1.0 that is local to this autonomous system.

```
BGP table version is 12, local router ID is 10.1.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
```

Troubleshooting Tips

Use the **ping** command to check basic network connectivity between the BGP routers.

Configuring a BGP Peer

Perform this task to configure BGP between two IPv4 routers (peers). The address family configured here is the default IPv4 unicast address family and the configuration is done at Router A in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

Before you begin

Before you perform this task, perform the "Configuring a BGP Routing Process" task shown in the prior section.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, such as IPv6 prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **neighbor** *ip-address* **activate**
7. **end**
8. **show ip bgp** [*network*] [*network-mask*]
9. **show ip bgp neighbors** [*neighbor-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

	Command or Action	Purpose
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.
Step 7	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 8	**show ip bgp** [*network*] [*network-mask*] **Example:** `Router# show ip bgp`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 9	**show ip bgp neighbors** [*neighbor-address*] **Example:** `Router(config-router-af)# show ip bgp neighbors 192.168.2.2`	(Optional) Displays information about the TCP and BGP connections to neighbors. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output from the **show ip bgp** command shows the BGP routing table for Router A in the figure above after this task has been configured on Router A and Router B. You can now see an entry for the network 172.17.1.0 in autonomous system 45000.

```
BGP table version is 13, local router ID is 10.1.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*> 172.17.1.0/24    192.168.1.1              0             0 45000 i
```

The following sample output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.1.1 of Router A in the figure above after this task has been configured on Router A:

```
BGP neighbor is 192.168.1.1,  remote AS 45000, external link
  BGP version 4, remote router ID 172.17.1.99
  BGP state = Established, up for 00:06:55
  Last read 00:00:15, last write 00:00:15, hold time is 120, keepalive intervals
  Configured hold time is 120,keepalive interval is 70 seconds, Minimum holdtims
  Neighbor capabilities:
    Route refresh: advertised and received (old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            13         13
    Route Refresh:          0          0
    Total:                 15         16
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 13, neighbor version 13/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:             1          1 (Consumes 52 bytes)
    Prefixes Total:               1          1
    Implicit Withdraw:            0          0
    Explicit Withdraw:            0          0
    Used as bestpath:           n/a          1
    Used as multipath:          n/a          0
                                  Outbound    Inbound
  Local Policy Denied Prefixes:   --------    -------
    AS_PATH loop:                      n/a          1
    Bestpath from this peer:             1        n/a
    Total:                               1          1
  Number of NLRIs in the update sent: max 0, min 0
  Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 192.168.1.2, Local port: 179
Foreign host: 192.168.1.1, Foreign port: 37725
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x12F4F2C):
Timer          Starts    Wakeups            Next
Retrans            14          0             0x0
TimeWait            0          0             0x0
AckHold            13          8             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
```

```
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss:  165379618  snduna:  165379963  sndnxt:  165379963     sndwnd:  16040
irs: 3127821601  rcvnxt: 3127821993  rcvwnd:      15993  delrcvwnd:    391
SRTT: 254 ms, RTTO: 619 ms, RTV: 365 ms, KRTT: 0 ms
minRTT: 12 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
IP Precedence value : 6
Datagrams (max data segment is 1460 bytes):
Rcvd: 20 (out of order: 0), with data: 15, total data bytes: 391
Sent: 22 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 04
```

Troubleshooting Tips

Use the **ping** command to verify basic network connectivity between the BGP routers.

Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Perform this task to configure a Border Gateway Protocol (BGP) routing process and BGP peers when the BGP peers are located in an autonomous system (AS) that uses 4-byte AS numbers. The address family configured here is the default IPv4 unicast address family, and the configuration is done at Router B in the figure above (in the "Cisco Implementation of 4-Byte Autonomous System Numbers" section). The 4-byte AS numbers in this task are formatted in the default asplain (decimal value) format; for example, Router B is in AS number 65538 in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

Before you begin

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. Repeat Step 4 to define other BGP neighbors, as required.
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **neighbor** *ip-address* **activate**
8. Repeat Step 7 to activate other BGP neighbors, as required.
9. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
10. **end**
11. **show ip bgp** [*network*] [*network-mask*]
12. **show ip bgp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 65536`	Adds the IP address of the neighbor in the specified AS to the IPv4 multiprotocol BGP neighbor table of the local device. • In this example, the 4-byte AS number, 65536, is defined in asplain notation.
Step 5	Repeat Step 4 to define other BGP neighbors, as required.	--
Step 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the device is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local device.
Step 8	Repeat Step 7 to activate other BGP neighbors, as required.	--

	Command or Action	Purpose
Step 9	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this AS and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp 10.1.1.0`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 12	**show ip bgp summary** **Example:** `Device# show ip bgp summary`	(Optional) Displays the status of all BGP connections.

Examples

The following output from the **show ip bgp** command at Router B shows the BGP routing table entry for network 10.1.1.0 learned from the BGP neighbor at 192.168.1.2 in Router A in the figure above with its 4-byte AS number of 65536 displayed in the default asplain format.

```
RouterB# show ip bgp 10.1.1.0

BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1)
  Advertised to update-groups:
     2
  65536
    192.168.1.2 from 192.168.1.2 (10.1.1.99)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

The following output from the **show ip bgp summary** command shows the 4-byte AS number 65536 for the BGP neighbor 192.168.1.2 of Router A in the figure above after this task has been configured on Router B:

```
RouterB# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
3/2 BGP path/bestpath attribute entries using 444 bytes of memory
```

```
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 806 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  Stated
192.168.1.2     4       65536       6       6        3    0    0 00:01:33       1
```

Troubleshooting Tips

Use the **ping** command to verify basic network connectivity between the BGP routers.

Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers

Perform this task to modify the default output format for 4-byte autonomous system (AS) numbers from asplain format to asdot notation format. The **show ip bgp summary** command is used to display the changes in output format for the 4-byte AS numbers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp summary**
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **bgp asnotation dot**
6. **end**
7. **clear ip bgp ***
8. **show ip bgp summary**
9. **show ip bgp regexp** *regexp*
10. **configure terminal**
11. **router bgp** *autonomous-system-number*
12. **no bgp asnotation dot**
13. **end**
14. **clear ip bgp ***

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp summary** **Example:** `Device# show ip bgp summary`	Displays the status of all Border Gateway Protocol (BGP) connections.

	Command or Action	Purpose
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 5	**bgp asnotation dot** **Example:** `Device(config-router)# bgp asnotation dot`	Changes the default output format of BGP 4-byte AS numbers from asplain (decimal values) to dot notation. **Note** 4-byte AS numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for **show** commands or the matching of regular expressions.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 7	**clear ip bgp *** **Example:** `Device# clear ip bgp *`	Clears and resets all current BGP sessions. • In this example, a hard reset is performed to ensure that the 4-byte AS number format change is reflected in all BGP sessions. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 8	**show ip bgp summary** **Example:** `Device# show ip bgp summary`	Displays the status of all BGP connections.
Step 9	**show ip bgp regexp** *regexp* **Example:** `Device# show ip bgp regexp ^1\.0$`	Displays routes that match the AS path regular expression. • In this example, a regular expression to match a 4-byte AS path is configured using asdot format.
Step 10	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 11	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 12	**no bgp asnotation dot** **Example:** `Device(config-router)# no bgp asnotation dot`	Resets the default output format of BGP 4-byte AS numbers back to asplain (decimal values). **Note** 4-byte AS numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for **show** commands or the matching of regular expressions.
Step 13	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 14	**clear ip bgp *** **Example:** `Device# clear ip bgp *`	Clears and resets all current BGP sessions. • In this example, a hard reset is performed to ensure that the 4-byte AS number format change is reflected in all BGP sessions. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following output from the **show ip bgp summary** command shows the default asplain format of the 4-byte AS numbers. Note the asplain format of the 4-byte AS numbers, 65536 and 65550.

```
Router# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4        65536       7       7        1    0    0 00:03:04      0
192.168.3.2     4        65550       4       4        1    0    0 00:00:15      0
```

After the **bgp asnotation dot** command is configured (followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions), the output is converted to asdot notation format as shown in the following output from the **show ip bgp summary** command. Note the asdot format of the 4-byte AS numbers, 1.0 and 1.14 (these are the asdot conversions of the 65536 and 65550 AS numbers.

```
Router# show ip bgp summary
```

```
BGP router identifier 172.17.1.99, local AS number 1.2
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4         1.0       9       9        1    0    0 00:04:13      0
192.168.3.2     4        1.14       6       6        1    0    0 00:01:24      0
```

After the **bgp asnotation dot** command is configured (followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions), the regular expression match format for 4-byte AS paths is changed to asdot notation format. Although a 4-byte AS number can be configured in a regular expression using either asplain format or asdot format, only 4-byte AS numbers configured using the current default format are matched. In the first example below, the **show ip bgp regexp** command is configured with a 4-byte AS number in asplain format. The match fails because the default format is currently asdot format and there is no output. In the second example using asdot format, the match passes and the information about the 4-byte AS path is shown using the asdot notation.

Note The asdot notation uses a period, which is a special character in Cisco regular expressions. To remove the special meaning, use a backslash before the period.

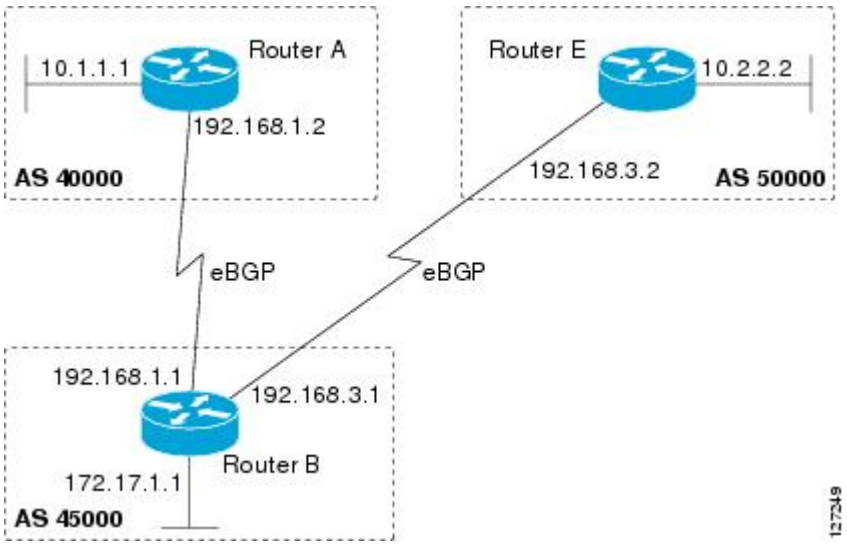
```
Router# show ip bgp regexp ^65536$

Router# show ip bgp regexp ^1\.0$

BGP table version is 2, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 1.0 i
```

Configuring a BGP Peer for the IPv4 VRF Address Family

Perform this optional task to configure BGP between two IPv4 routers (peers) that must exchange IPv4 VRF information because they exist in a VPN. The address family configured here is the IPv4 VRF address family and the configuration is done at Router B in the figure below with the neighbor 192.168.3.2 at Router E in autonomous system 50000. Remember to perform this task for any neighbor routers that are to be BGP IPv4 VRF address family peers.

This task does not show the complete configuration required for VPN routing. For some complete example configurations and an example configuration showing how to create a VRF with a route-target that uses a 4-byte autonomous system number, see .

Figure 43: BGP Topology for IPv4 VRF Address Family



Before you begin

Before you perform this task, perform the Configuring a BGP Routing Process, on page 525task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **exit**
7. **router bgp** *autonomous-system-number*
8. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
9. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
10. **neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*] [**restart** *restart-interval*] [**warning-only**]
11. **neighbor** *ip-address* **activate**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Configures a VRF routing table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 45000:5`	Creates routing and forwarding tables and specifies the default route distinguisher for a VPN. • Use the *route-distinguisher* argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target both 45000:100`	Creates a route target extended community for a VRF. • Use the **import** keyword to import routing information from the target VPN extended community. • Use the **export** keyword to export routing information to the target VPN extended community. • Use the **both** keyword to import both import and export routing information to the target VPN extended community. • Use the *route-target-ext-community* argument to add the route target extended community attributes to the VRF's list of import, export, or both (import and export) route target extended communities.
Step 6	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 8	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:**	Specifies the IPv4 address family and enters address family configuration mode. • Use the **unicast** keyword to specify the IPv4 unicast address family. By default, the router is placed in

	Command or Action	Purpose
	`Router(config-router)# address-family ipv4 vrf vpn1`	configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 9	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.3.2 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*] [**restart** *restart-interval*] [**warning-only**] **Example:** `Router(config-router-af)# neighbor 192.168.3.2 maximum-prefix 10000 warning-only`	Controls how many prefixes can be received from a neighbor. • Use the *maximum* argument to specify the maximum number of prefixes allowed from the specified neighbor. The number of prefixes that can be configured is limited only by the available system resources on a router. • Use the *threshold* argument to specify an integer representing a percentage of the maximum prefix limit at which the router starts to generate a warning message. • Use the **warning-only** keyword to allow the router to generate a log message when the maximum prefix limit is exceeded, instead of terminating the peering session.
Step 11	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.3.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.
Step 12	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

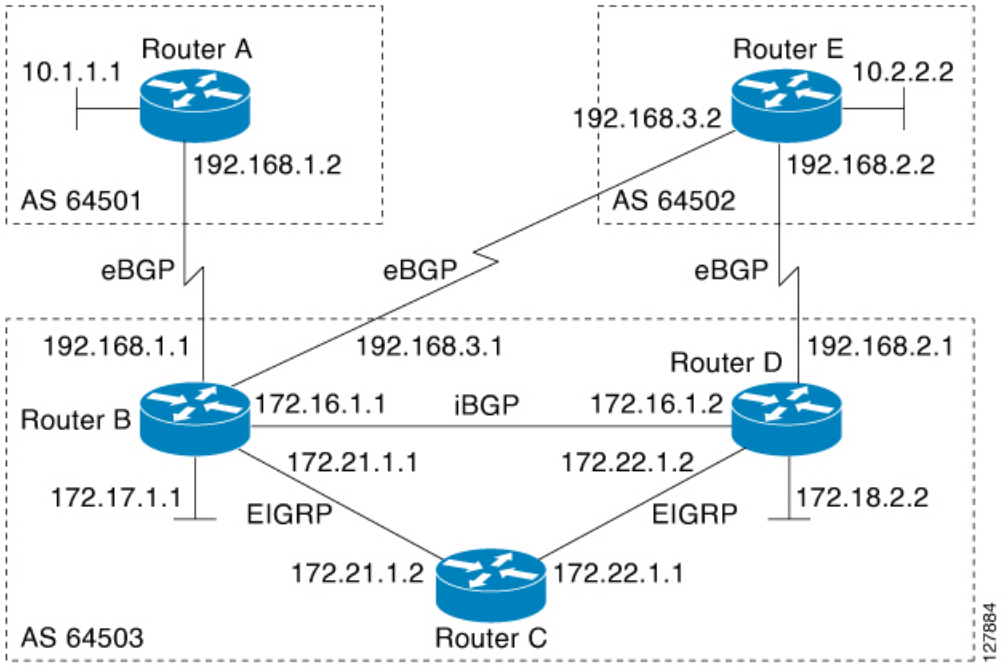
Use the **ping** command to verify basic network connectivity between the BGP routers, and use the **show ip vrf** command to verify that the VRF instance has been created.

Customizing a BGP Peer

Perform this task to customize your BGP peers. Although many of the steps in this task are optional, this task demonstrates how the neighbor and address family configuration command relationships work. Using the example of the IPv4 multicast address family, neighbor address family-independent commands are configured before the IPv4 multicast address family is configured. Commands that are address family-dependent are then configured and the **exit address-family** command is shown. An optional step shows how to disable a neighbor.

The configuration in this task is done at Router B in the figure below and would need to be repeated with appropriate changes to the IP addresses, for example, at Router E to fully configure a BGP process between the two devices.

Figure 44: BGP Peer Topology



Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, such as IPv6 prefixes.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **description** *text*
7. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
8. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
9. **neighbor** {*ip-address* | *peer-group-name*} **activate**
10. **neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds*
11. **neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]
12. **exit-address-family**
13. **neighbor** {*ip-address* | *peer-group-name*} **shutdown**
14. **end**
15. **show ip bgp ipv4 multicast** [*command*]
16. **show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received prefix-filter**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast** router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.3.2 remote-as 50000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **description** *text* **Example:** `Device(config-router)# neighbor 192.168.3.2 description finance`	(Optional) Associates a text description with the specified neighbor.
Step 7	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 multicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the device is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 8	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.3.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds* **Example:**	(Optional) Sets the minimum interval between the sending of BGP routing updates.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.3.2 advertisement-interval 25`	
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*] **Example:** `Device(config-router-af)# neighbor 192.168.3.2 default-originate`	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 12	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 13	**neighbor** {*ip-address* | *peer-group-name*} **shutdown** **Example:** `Device(config-router)# neighbor 192.168.3.2 shutdown`	(Optional) Disables a BGP peer or peer group. **Note** If you perform this step you will not be able to run either of the subsequent **show** command steps because you have disabled the neighbor.
Step 14	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 15	**show ip bgp ipv4 multicast** [*command*] **Example:** `Device# show ip bgp ipv4 multicast`	(Optional) Displays IPv4 multicast database-related information. • Use the *command* argument to specify any multiprotocol BGP command that is supported. To see the supported commands, use the ? prompt on the CLI.
Step 16	**show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received prefix-filter**] **Example:** `Device# show ip bgp neighbors 192.168.3.2`	(Optional) Displays information about the TCP and BGP connections to neighbors.

Examples

The following sample output from the **show ip bgp ipv4 multicast** command shows BGP IPv4 multicast information for Router B in the figure above after this task has been configured on Router B and Router E. Note that the networks local to each device that were configured under IPv4 multicast address family appear in the output table.

```
BGP table version is 3, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.3.2              0             0 50000 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

The following partial sample output from the **show ip bgp neighbors** command for neighbor 192.168.3.2 shows general BGP information and specific BGP IPv4 multicast address family information about the neighbor. The command was entered on Router B in the figure above after this task had been configured on Router B and Router E.

```
BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Description: finance
  BGP version 4, remote router ID 10.2.2.99
  BGP state = Established, up for 01:48:27
  Last read 00:00:26, last write 00:00:26, hold time is 120, keepalive intervals
  Configured hold time is 120,keepalive interval is 70 seconds, Minimum holdtims
  Neighbor capabilities:
    Route refresh: advertised and received (old & new)
    Address family IPv4 Unicast: advertised
    Address family IPv4 Multicast: advertised and received
!
 For address family: IPv4 Multicast
  BGP table version 3, neighbor version 3/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
    Uses NEXT_HOP attribute for MBGP NLRIs
                                 Sent       Rcvd
  Prefix activity:               ----       ----
    Prefixes Current:               1          1 (Consumes 48 bytes)
    Prefixes Total:                 1          1
    Implicit Withdraw:              0          0
    Explicit Withdraw:              0          0
    Used as bestpath:             n/a          1
    Used as multipath:            n/a          0
                                   Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    Bestpath from this peer:                1        n/a
    Total:                                  1          0
  Number of NLRIs in the update sent: max 0, min 0
  Minimum time between advertisement runs is 25 seconds
  Connections established 8; dropped 7
  Last reset 01:48:54, due to User reset
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 192.168.3.1, Local port: 13172
Foreign host: 192.168.3.2, Foreign port: 179
!
```

Removing BGP Configuration Commands Using a Redistribution

BGP CLI configuration can become quite complex even in smaller BGP networks. If you need to remove any CLI configuration, you must consider all the implications of removing the CLI. Analyze the current running configuration to determine the current BGP neighbor relationships, any address family considerations, and even other routing protocols that are configured. Many BGP CLI commands affect other parts of the CLI configuration.

Perform this task to remove all the BGP configuration commands used in a redistribution of BGP routes into EIGRP. A route map can be used to match and set parameters or to filter the redistributed routes to ensure that routing loops are not created when these routes are subsequently advertised by EIGRP. When removing BGP configuration commands you must remember to remove or disable all the related commands. In this example, if the **route-map** command is omitted, then the redistribution will still occur and possibly with unexpected results as the route map filtering has been removed. Omitting just the **redistribute** command would mean that the route map is not applied, but it would leave unused commands in the running configuration.

For more details on BGP CLI removal, see the "BGP CLI Removal Considerations" concept in the "Cisco BGP Overview" module.

To view the redistribution configuration before and after the CLI removal, see the "Examples: Removing BGP Configuration Commands Using a Redistribution Example" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no route-map** *map-name*
4. **router eigrp** *autonomous-system-number*
5. **no redistribute** *protocol* [*as-number*]
6. **end**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no route-map** *map-name* **Example:** `Device(config)# no route-map bgp-to-eigrp`	Removes a route map from the running configuration. • In this example, a route map named bgp-to-eigrp is removed from the configuration.
Step 4	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 100`	Enters router configuration mode for the specified routing process.
Step 5	**no redistribute** *protocol* [*as-number*] **Example:**	Disables the redistribution of routes from one routing domain into another routing domain.

	Command or Action	Purpose
	`Device(config-router)# no redistribute bgp 45000`	• In this example, the configuration of the redistribution of BGP routes into the EIGRP routing process is removed from the running configuration. **Note** If a route map was included in the original **redistribute** command configuration, remember to remove the **route-map** command configuration as in Step 3 in this example task. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	(Optional) Displays the current running configuration on the router. • Use this command to verify that the **redistribute** and **route-map** commands are removed from the router configuration.

Monitoring and Maintaining Basic BGP

The tasks in this section are concerned with the resetting and display of information about basic BGP processes and peer relationships. Once you have defined two routers to be BGP neighbors, they will form a BGP connection and exchange routing information. If you subsequently change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you may have to reset BGP connections for the configuration change to take effect.

Configuring Inbound Soft Reconfiguration When Route Refresh Capability Is Missing

Perform this task to configure inbound soft reconfiguration using the **bgp soft-reconfig-backup** command for BGP peers that do not support the route refresh capability. BGP peers that support the route refresh capability are unaffected by the configuration of this command. Note that the memory requirements for storing the inbound update information can become quite large.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp log-neighbor-changes**
5. **bgp soft-reconfig-backup**
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*

7. **neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [**inbound**]
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.
10. **exit**
11. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address*
13. **end**
14. **show ip bgp neighbors** [*neighbor-address*]
15. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**bgp soft-reconfig-backup** **Example:** `Device(config-router)# bgp soft-reconfig-backup`	Configures a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability. • This command is used to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:**	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.

	Command or Action	Purpose
	`Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [**inbound**] **Example:** `Device(config-router)# neighbor 192.168.1.2 soft-reconfiguration inbound`	Configures the Cisco software to start storing updates. • All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is done later, the stored information will be used to generate a new set of inbound updates.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router)# neighbor 192.168.1.2 route-map LOCAL in`	Applies a route map to incoming or outgoing routes. • In this example, the route map named LOCAL will be applied to incoming routes.
Step 9	Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.	—
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 11	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map LOCAL permit 10`	Configures a route map and enters route-map configuration mode. • In this example, a route map named LOCAL is created.
Step 12	**set ip next-hop** *ip-address* **Example:** `Device(config-route-map)# set ip next-hop 192.168.1.144`	Specifies where output packets that pass a match clause of a route map for policy routing. • In this example, the ip address is set to 192.168.1.144.
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	**show ip bgp neighbors** [*neighbor-address*] **Example:** `Device# show ip bgp neighbors 192.168.1.2`	(Optional) Displays information about the TCP and BGP connections to neighbors. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

	Command or Action	Purpose
Step 15	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following partial output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.2.1. This peer supports route refresh.

```
BGP neighbor is 192.168.1.2,  remote AS 40000, external link
 Neighbor capabilities:
   Route refresh: advertised and received(new)
```

The following partial output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.3.2. This peer does not support route refresh so the soft-reconfig inbound paths for BGP peer 192.168.3.2 will be stored because there is no other way to update any inbound policy updates.

```
BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Neighbor capabilities:
   Route refresh: advertised
```

The following sample output from the **show ip bgp** command shows the entry for the network 172.17.1.0. Both BGP peers are advertising 172.17.1.0/24, but only the received-only path is stored for 192.168.3.2.

```
BGP routing table entry for 172.17.1.0/24, version 11
Paths: (3 available, best #3, table Default-IP-Routing-Table, RIB-failure(4))
Flag: 0x820
 Advertised to update-groups:
    1
 50000
   192.168.3.2 from 192.168.3.2 (172.17.1.0)
     Origin incomplete, metric 0, localpref 200, valid, external
 50000, (received-only)
   192.168.3.2 from 192.168.3.2 (172.17.1.0)
     Origin incomplete, metric 0, localpref 100, valid, external
 40000
   192.168.1.2 from 192.168.1.2 (172.16.1.0)
     Origin incomplete, metric 0, localpref 200, valid, external, best
```

Resetting and Displaying Basic BGP Information

Perform this task to reset and display information about basic BGP processes and peer relationships.

SUMMARY STEPS

1. **enable**
2. **clear ip bgp** {***** | *autonomous-system-number* | *neighbor-address*} [**soft** [**in** | **out**]]

3. **show ip bgp** [*network-address*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter prefixes** *mask-length*]
4. **show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*]
5. **show ip bgp paths**
6. **show ip bgp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp** {***** | *autonomous-system-number* | *neighbor-address*} [**soft** [**in** | **out**]] **Example:** `Device# clear ip bgp *`	Clears and resets BGP neighbor sessions: • In the example provided, all BGP neighbor sessions are cleared and reset.
Step 3	**show ip bgp** [*network-address*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter prefixes** *mask-length*] **Example:** `Device# show ip bgp 10.1.1.0 255.255.255.0`	Displays all the entries in the BGP routing table: • In the example provided, the BGP routing table information for the 10.1.1.0 network is displayed.
Step 4	**show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*] **Example:** `Device# show ip bgp neighbors 192.168.3.2 advertised-routes`	Displays information about the TCP and BGP connections to neighbors. • In the example provided, the routes advertised from the device to BGP neighbor 192.168.3.2 on another device are displayed.
Step 5	**show ip bgp paths** **Example:** `Device# show ip bgp paths`	Displays information about all the BGP paths in the database.
Step 6	**show ip bgp summary** **Example:** `Device# show ip bgp summary`	Displays information about the status of all BGP connections.

Aggregating Route Prefixes Using BGP

BGP peers exchange information about local networks, but this can quickly lead to large BGP routing tables. CIDR enables the creation of aggregate routes (or *supernets*) to minimize the size of routing tables. Smaller BGP routing tables can reduce the convergence time of the network and improve network performance. Aggregated routes can be configured and advertised using BGP. Some aggregations advertise only summary routes and other methods of aggregating routes allow more specific routes to be forwarded. Aggregation applies only to routes that exist in the BGP routing table. An aggregated route is forwarded if at least one more specific route of the aggregation exists in the BGP routing table. Perform one of the following tasks to aggregate routes within BGP:

Redistributing a Static Aggregate Route into BGP

Use this task to redistribute a static aggregate route into BPG. A static aggregate route is configured and then redistributed into the BGP routing table. The static route must be configured to point to interface null 0 and the prefix should be a superset of known BGP routes. When a device receives a BGP packet, it will use the more specific BGP routes. If the route is not found in the BGP routing table, then the packet will be forwarded to null 0 and discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
4. **router bgp** *autonomous-system-number*
5. **redistribute static**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*] **Example:** `Device(config)# ip route 172.0.0.0 255.0.0.0 null 0`	Creates a static route.

	Command or Action	Purpose
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 5	**redistribute static** **Example:** `Device(config-router)# redistribute static`	Redistributes routes into the BGP routing table.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Conditional Aggregate Routes Using BGP

Use this task to create an aggregate route entry in the BGP routing table when at least one specific route falls into the specified range. The aggregate route is advertised as originating from your autonomous system. For more information, see the "BGP Route Aggregation Generating AS_SET Information" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **aggregate-address** *address mask* [**as-set**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	**aggregate-address** *address mask* [**as-set**] **Example:** `Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 as-set`	Creates an aggregate entry in a BGP routing table. • A specified route must exist in the BGP table. • Use the **aggregate-address** command with no keywords to create an aggregate entry if any more-specific BGP routes are available that fall in the specified range. • Use the **as-set** keyword to specify that the path advertised for this route is an AS_SET. Do not use the **as-set** keyword when aggregating many paths because this route is withdrawn and updated every time the reachability information for the aggregated route changes. **Note** Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Suppressing and Unsuppressing the Advertisement of Aggregated Routes Using BGP

Use this task to create an aggregate route, suppress the advertisement of routes using BGP, and subsequently unsuppress the advertisement of routes. Routes that are suppressed are not advertised to any neighbors, but it is possible to unsuppress routes that were previously suppressed to specific neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. Do one of the following:
   - **aggregate-address** *address mask* [**summary-only**]
   - **aggregate-address** *address mask* [**suppress-map** *map-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **unsuppress-map** *map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.

Step 5	Do one of the following:
	• **aggregate-address** *address mask* [**summary-only**]
	• **aggregate-address** *address mask* [**suppress-map** *map-name*]
	Example:
	`Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 summary-only`
	Example:
	`Device(config-router)# aggregate-address 172.0.0.0 255.0.0.0 suppress-map map1`
		Creates an aggregate route.
		• Use the optional **summary-only** keyword to create the aggregate route (for example, 10.*.*.*) and also suppresses advertisements of more-specific routes to all neighbors.
		• Use the optional **suppress-map** keyword to create the aggregate route but suppress advertisement of specified routes. Routes that are suppressed are not advertised to any neighbors. You can use the **match** clauses of route maps to selectively suppress some more-specific routes of the aggregate and leave others unsuppressed. IP access lists and autonomous system path access lists **match** clauses are supported.
		Note Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Step 6	**neighbor** {*ip-address* | *peer-group-name*} **unsuppress-map** *map-name* **Example:** `Device(config-router)# neighbor 192.168.1.2 unsuppress map1`	(Optional) Selectively advertises routes previously suppressed by the **aggregate-address** command. • In this example, the routes previously suppressed in Step 5 are advertised to neighbor 192.168.1.2.
Step 7	**end** **Example:**	Exits router configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router)# end`	

Suppressing Inactive Route Advertisement Using BGP

Perform this task to suppress the advertisement of inactive routes by BGP. In Cisco IOS Release 12.2(25)S, 12.2(33)SXH, and 15.0(1)M, the **bgp suppress-inactive** command was introduced to configure BGP to not advertise inactive routes to any BGP peer. A BGP routing process can advertise routes that are not installed in the RIB to BGP peers by default. A route that is not installed into the RIB is an inactive route. Inactive route advertisement can occur, for example, when routes are advertised through common route aggregation.

Inactive route advertisements can be suppressed to provide more consistent data forwarding. This feature can be configured on a per IPv4 address family basis. For example, when specifying the maximum number of routes that can be configured in a VRF with the **maximum routes** global configuration command, you also suppress inactive route advertisement to prevent inactive routes from being accepted into the VRF after route limit has been exceeded.

Before you begin

This task assumes that BGP is enabled and that peering has been established.

Note Inactive route suppression can be configured only under the IPv4 address family or under a default IPv4 general session.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
5. **bgp suppress-inactive**
6. **end**
7. **show ip bgp rib-failure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enter address family configuration mode to configure BGP peers to accept address family specific configurations. • The example creates an IPv4 unicast address family session.
Step 5	**bgp suppress-inactive** **Example:** `Router(config-router-af)# bgp suppress-inactive`	Suppresses BGP advertising of inactive routes. • BGP advertises inactive routes by default. • Entering the **no** form of this command reenables the advertisement of inactive routes.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 7	**show ip bgp rib-failure** **Example:** `Router# show ip bgp rib-failure`	(Optional) Displays BGP routes that are not installed in the RIB.

Examples

The following example shows output from the **show ip bgp rib-failure** command displaying routes that are not installed in the RIB. The output shows that the displayed routes were not installed because a route or routes with a better administrative distance already exist in the RIB.

```
Router# show ip bgp rib-failure

Network            Next Hop                    RIB-failure   RIB-NH Matches
10.1.15.0/24       10.1.35.5          Higher admin distance             n/a
10.1.16.0/24       10.1.15.1          Higher admin distance             n/a
```

Conditionally Advertising BGP Routes

Perform this task to conditionally advertise selected BGP routes. The routes or prefixes that will be conditionally advertised are defined in two route maps: an advertise map and either an exist map or nonexist map. The route map associated with the exist map or nonexist map specifies the prefix that the BGP speaker will track. The

route map associated with the advertise map specifies the prefix that will be advertised to the specified neighbor when the condition is met.

- If a prefix is found to be present in the exist map by the BGP speaker, the prefix specified by the advertise map is advertised.
- If a prefix is found not to be present in the nonexist map by the BGP speaker, the prefix specified by the advertise map is advertised.

If the condition is not met, the route is withdrawn and conditional advertisement does not occur. All routes that may be dynamically advertised or not advertised must exist in the BGP routing table in order for conditional advertisement to occur. These routes are referenced from an access list or an IP prefix list. Note, when configuring an advertise-map, ensure that BGP attributes are set within the advertise-map and not in a separate route-map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **advertise-map** *map-name* {**exist-map** *map-name* | **non-exist-map** *map-name*}
6. **exit**
7. **route-map** *map-tag* [**permit** | **deny**] [**sequence-number**]
8. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
12. **exit**
13. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
14. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 5	**neighbor** *ip-address* **advertise-map** *map-name* {**exist-map** *map-name* | **non-exist-map** *map-name*} **Example:** `Device(config-router)# neighbor 192.168.1.2 advertise-map map1 exist-map map2`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. • In this example, the prefix (172.17.0.0) matching the ACL in the advertise map (the route map named map1) will be advertised to the neighbor only when a prefix (192.168.50.0) matching the ACL in exist map (the route-map named map2) is in the local BGP table.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 7	**route-map** *map-tag* [**permit** | **deny**] [**sequence-number**] **Example:** `Device(config)# route-map map1 permit 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named map1 is created.
Step 8	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Device(config-route-map)# match ip address 1`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match a prefix permitted by access list 1.
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:**	Configures a route map and enters route map configuration mode.

	Command or Action	Purpose
	`Device(config)# route-map map2 permit 10`	• In this example, a route map named map2 is created.
Step 11	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Device(config-route-map)# match ip address 2`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match a prefix permitted by access list 2.
Step 12	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 13	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Device(config)# access-list 1 permit 172.17.0.0`	Configures a standard access list. • In this example, access list 1 permits advertising of the 172.17.0.0 prefix, depending on other conditions set by the **neighbor advertise-map** command.
Step 14	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Device(config)# access-list 2 permit 192.168.50.0`	Configures a standard access list. • In this example, access list 2 permits the 192.168.50.0 to be the prefix of the exist-map.
Step 15	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Originating BGP Routes

Route aggregation is useful to minimize the size of the BGP table, but there are situations when you want to add more specific prefixes to the BGP table. Route aggregation can hide more specific routes. Using the **network** command as shown in the "Configuring a BGP Routing Process" section originates routes, and the following optional tasks originate BGP routes for the BGP table for different situations.

Advertising a Default Route Using BGP

Perform this task to advertise a default route to BGP peers. The default route is locally originated. A default route can be useful to simplify configuration or to prevent the device from using too many system resources. If the device is peered with an Internet service provider (ISP), the ISP will carry full routing tables, so configuring a default route into the ISP network saves resources at the local device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network / length* | **permit** *network / length*} [**ge** *ge-value*] [**le** *le-value*]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
6. **exit**
7. **router bgp** *autonomous-system-number*
8. **neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network / length*	**permit** *network / length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Device(config)# ip prefix-list DEFAULT permit 10.1.1.0/24`	Configures an IP prefix list. • In this example, prefix list DEFAULT permits advertising of the 10.1.1.0/24. prefix depending on a match set by the **match ip address** command.
Step 4	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] **Example:** `Device(config)# route-map ROUTE`	Configures a route map and enters route map configuration mode. • In this example, a route map named ROUTE is created.
Step 5	**match ip address** {*access-list-number* [*access-list-number...*	*access-list-name...*]	*access-list-name* [*access-list-number...*	*access-list-name*]	**prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Device(config-route-map)# match ip address prefix-list DEFAULT`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match a prefix permitted by prefix list DEFAULT.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 7	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*] **Example:** `Device(config-router)# neighbor 192.168.3.2 default-originate`	(Optional) Permits a BGP speaker--the local device--to send the default route 0.0.0.0 to a peer for use as a default route.
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

Use the **show ip route** command on the receiving BGP peer (not on the local router) to verify that the default route has been set. In the output, verify that a line similar to the following showing the default route 0.0.0.0 is present:

```
B*   0.0.0.0/0 [20/0] via 192.168.1.2, 00:03:10
```

Conditionally Injecting BGP Routes

Use this task to inject more specific prefixes into a BGP routing table over less specific prefixes that were selected through normal route aggregation. These more specific prefixes can be used to provide a finer granularity of traffic engineering or administrative control than is possible with aggregated routes. For more information, see the “Conditional BGP Route Injection” section.

Before you begin

This task assumes that the IGP is already configured for the BGP peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**]
5. **exit**

6. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
7. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
8. **match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...*| *access-list-name...*]
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **set ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
12. **set community** {*community-number* [**additive**] [*well-known-community*] | **none**}
13. **exit**
14. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
15. Repeat Step 14 for every prefix list to be created.
16. **exit**
17. **show ip bgp injected-paths**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**] **Example:** `Router(config-router)# bgp inject-map ORIGINATE exist-map LEARNED_PATH`	Specifies the inject map and the exist map for conditional route injection. • Use the **copy-attributes** keyword to specify that the injected route inherit the attributes of the aggregate route.
Step 5	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 6	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map LEARNED_PATH permit 10`	Configures a route map and enters route map configuration mode.
Step 7	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Router(config-route-map)# match ip address prefix-list SOURCE`	Specifies the aggregate route to which a more specific route will be injected. • In this example, the prefix list named SOURCE is used to redistribute the source of the route.
Step 8	**match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...*| *access-list-name...*] **Example:** `Router(config-route-map)# match ip route-source prefix-list ROUTE_SOURCE`	Specifies the match conditions for redistributing the source of the route. • In this example, the prefix list named ROUTE_SOURCE is used to redistribute the source of the route. **Note** The route source is the neighbor address that is configured with the **neighbor remote-as** command. The tracked prefix must come from this neighbor in order for conditional route injection to occur.
Step 9	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map ORIGINATE permit 10`	Configures a route map and enters route map configuration mode.
Step 11	**set ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Router(config-route-map)# set ip address prefix-list ORIGINATED_ROUTES`	Specifies the routes to be injected. • In this example, the prefix list named originated_routes is used to redistribute the source of the route.
Step 12	**set community** {*community-number* [**additive**] [*well-known-community*] | **none**}	Sets the BGP community attribute of the injected route.

	Command or Action	Purpose
	Example: `Router(config-route-map)# set community 14616:555 additive`	
Step 13	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 14	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list SOURCE permit 10.1.1.0/24`	Configures a prefix list. • In this example, the prefix list named SOURCE is configured to permit routes from network 10.1.1.0/24.
Step 15	Repeat Step 14 for every prefix list to be created.	--
Step 16	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 17	**show ip bgp injected-paths** **Example:** `Router# show ip bgp injected-paths`	(Optional) Displays information about injected paths.

Examples

The following sample output is similar to the output that will be displayed when the **show ip bgp injected-paths**command is entered:

```
Router# show ip bgp injected-paths

BGP table version is 11, local router ID is 10.0.0.1
Status codes:s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes:i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       10.0.0.2                               0 ?
*> 172.17.0.0/16    10.0.0.2                               0 ?
```

Troubleshooting Tips

BGP conditional route injection is based on the injection of a more specific prefix into the BGP routing table when a less specific prefix is present. If conditional route injection is not working properly, verify the following:

- If conditional route injection is configured but does not occur, verify the existence of the aggregate prefix in the BGP routing table. The existence (or not) of the tracked prefix in the BGP routing table can be verified with the **show ip bgp**command.
- If the aggregate prefix exists but conditional route injection does not occur, verify that the aggregate prefix is being received from the correct neighbor and the prefix list identifying that neighbor is a /32 match.
- Verify the injection (or not) of the more specific prefix using the **show ip bgp injected-paths**command.
- Verify that the prefix that is being injected is not outside of the scope of the aggregate prefix.
- Ensure that the inject route map is configured with the **set ip address** command and not the **match ip address** command.

Originating BGP Routes Using Backdoor Routes

Use this task to indicate to border devices which networks are reachable using a backdoor route. A backdoor network is treated the same as a local network, except that it is not advertised. For more information, see the BGP Backdoor Routes section.

Before you begin

This task assumes that the IGP (EIGRP, in this example) is already configured for the BGP peers. The configuration is done at Router B in the in the "BGP Backdoor Routes" section, and the BGP peer is Router D.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **network** *ip-address* **backdoor**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:**	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 45000`	
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 172.22.1.2 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device. • In this example, the peer is an internal peer as the autonomous system number specified for the peer is the same number specified in Step 3.
Step 5	**network** *ip-address* **backdoor** **Example:** `Device(config-router)# network 172.21.1.0 backdoor`	Indicates a network that is reachable through a backdoor route.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring a BGP Peer Group

This task explains how to configure a BGP peer group. Often, in a BGP speaker, many neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into peer groups to simplify configuration and, more importantly, to make updating more efficient. When you have many peers, this approach is highly recommended.

The three steps to configure a BGP peer group, described in the following task, are as follows:

- Creating the peer group
- Assigning options to the peer group
- Making neighbors members of the peer group

You can disable a BGP peer or peer group without removing all the configuration information using the **neighbor shutdown** router configuration command.

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **peer-group** *peer-group-name*
7. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
8. **neighbor** *peer-group-name* **activate**
9. **neighbor** *ip-address* **peer-group** *peer-group-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router)# neighbor fingroup peer-group`	Creates a BGP peer group.
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device.
Step 6	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router)# neighbor 192.168.1.1 peer-group fingroup`	Assigns the IP address of a BGP neighbor to a peer group.

	Command or Action	Purpose
Step 7	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 multicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. This is the default. • The **multicast** keyword specifies that IPv4 multicast address prefixes will be exchanged. • The **vrf** keyword and *vrf-name* argument specify that IPv4 VRF instance information will be exchanged.
Step 8	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor fingroup activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local device. **Note** By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. To allow BGP to exchange other address prefix types, such as multicast that is configured in this example, neighbors must also be activated using the **neighbor activate** command.
Step 9	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 peer-group fingroup`	Assigns the IP address of a BGP neighbor to a peer group.
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring Peer Session Templates

The following tasks create and configure a peer session template:

Configuring a Basic Peer Session Template

Perform this task to create a basic peer session template with general BGP routing session commands that can be applied to many neighbors using one of the next two tasks.

Note The commands in Step 5 and 6 are optional and could be replaced with any supported general session commands.

Note The following restrictions apply to the peer session templates:

- A peer session template can directly inherit only one session template, and each inherited session template can also contain one indirectly inherited session template. So, a neighbor or neighbor group can be configured with only one directly applied peer session template and seven additional indirectly inherited peer session templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **remote-as** *autonomous-system-number*
6. **timers** *keepalive-interval hold-time*
7. **end**
8. **show ip bgp template peer-session** [*session-template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Router(config-router)# template peer-session INTERNAL-BGP`	Enters session-template configuration mode and creates a peer session template.
Step 5	**remote-as** *autonomous-system-number* **Example:**	(Optional) Configures peering with a remote neighbor in the specified autonomous system.

	Command or Action	Purpose
	`Router(config-router-stmp)# remote-as 202`	**Note** Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.
Step 6	**timers** *keepalive-interval hold-time* **Example:** `Router(config-router-stmp)# timers 30 300`	(Optional) Configures BGP keepalive and hold timers. • The hold time must be at least twice the keepalive time. **Note** Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.
Step 7	**end** **Example:** `Router(config-router)# end`	Exits session-template configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp template peer-session** [*session-template-name*] **Example:** `Router# show ip bgp template peer-session`	Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template with the *session-template-name* argument. This command also supports all standard output modifiers.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.

Configuring Peer Session Template Inheritance with the inherit peer-session Command

This task configures peer session template inheritance with the **inherit peer-session** command. It creates and configures a peer session template and allows it to inherit a configuration from another peer session template.

Note The commands in Steps 5 and 6 are optional and could be replaced with any supported general session commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **description** *text-string*
6. **update-source** *interface-type interface-number*
7. **inherit peer-session** *session-template-name*

8. **end**
9. **show ip bgp template peer-session** [*session-template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Router(config-router)# template peer-session CORE1`	Enter session-template configuration mode and creates a peer session template.
Step 5	**description** *text-string* **Example:** `Router(config-router-stmp)# description CORE-123`	(Optional) Configures a description. • The text string can be up to 80 characters. **Note** Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.
Step 6	**update-source** *interface-type interface-number* **Example:** `Router(config-router-stmp)# update-source loopback 1`	(Optional) Configures a router to select a specific source or interface to receive routing table updates. • The example uses a loopback interface. The advantage to this configuration is that the loopback interface is not as susceptible to the effects of a flapping interface. **Note** Any supported general session command can be used here. For a list of the supported commands, see the "Restrictions" section.
Step 7	**inherit peer-session** *session-template-name* **Example:**	Configures this peer session template to inherit the configuration of another peer session template. • The example configures this peer session template to inherit the configuration from INTERNAL-BGP. This

	Command or Action	Purpose
	`Router(config-router-stmp)# inherit peer-session INTERNAL-BGP`	template can be applied to a neighbor, and the configuration INTERNAL-BGP will be applied indirectly. No additional peer session templates can be directly applied. However, the directly inherited template can contain up to seven indirectly inherited peer session templates.
Step 8	**end** **Example:** `Router(config-router)# end`	Exits session-template configuration mode and enters privileged EXEC mode.
Step 9	**show ip bgp template peer-session** [*session-template-name*] **Example:** `Router# show ip bgp template peer-session`	Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template with the optional *session-template-name* argument. This command also supports all standard output modifiers.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited or applied by another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.

Configuring Peer Session Template Inheritance with the neighbor inherit peer-session Command

This task configures a router to send a peer session template to a neighbor to inherit the configuration from the specified peer session template with the **neighbor inherit peer-session** command. Use the following steps to send a peer session template configuration to a neighbor to inherit.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **inherit peer-session** *session-template-name*
6. **end**
7. **show ip bgp template peer-session** [*session-template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 172.16.0.1 remote-as 202`	Configures a peering session with the specified neighbor. • The explicit **remote-as** statement is required for the neighbor inherit statement in Step 5 to work. If a peering is not configured, the specified neighbor in Step 5 will not accept the session template.
Step 5	**neighbor** *ip-address* **inherit peer-session** *session-template-name* **Example:** `Router(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1`	Sends a peer session template to a neighbor so that the neighbor can inherit the configuration. • The example configures a router to send the peer session template named CORE1 to the 172.16.0.1 neighbor to inherit. This template can be applied to a neighbor, and if another peer session template is indirectly inherited in CORE1, the indirectly inherited configuration will also be applied. No additional peer session templates can be directly applied. However, the directly inherited template can also inherit up to seven additional indirectly inherited peer session templates.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 7	**show ip bgp template peer-session** [*session-template-name*] **Example:** `Router# show ip bgp template peer-session`	Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template with the optional *session-template-name* argument. This command also supports all standard output modifiers.

Configuring Peer Policy Templates

Configuring Basic Peer Policy Templates

Perform this task to create a basic peer policy template with BGP policy configuration commands that can be applied to many neighbors using one of the next two tasks.

Note The commands in Steps 5 through 7 are optional and could be replaced with any supported BGP policy configuration commands.

Note The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**]
6. **weight** *weight-value*
7. **prefix-list** *prefix-list-name* {**in** | **out**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:**	Enters router configuration mode and creates a BGP routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 45000`	
Step 4	**template peer-policy** *policy-template-name* **Example:** `Device(config-router)# template peer-policy GLOBAL`	Enters policy-template configuration mode and creates a peer policy template.
Step 5	**maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**] **Example:** `Device(config-router-ptmp)# maximum-prefix 10000`	(Optional) Configures the maximum number of prefixes that a neighbor will accept from this peer. **Note** Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the "Peer Policy Templates" section.
Step 6	**weight** *weight-value* **Example:** `Device(config-router-ptmp)# weight 300`	(Optional) Sets the default weight for routes that are sent from this neighbor. **Note** Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the "Peer Policy Templates" section.
Step 7	**prefix-list** *prefix-list-name* {**in** | **out**} **Example:** `Device(config-router-ptmp)# prefix-list NO-MARKETING in`	(Optional) Filters prefixes that are received by the router or sent from the router. • The prefix list in the example filters inbound internal addresses. **Note** Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the "Peer Policy Templates" section.
Step 8	**end** **Example:** `Device(config-router-ptmp)# end`	Exits policy-template configuration mode and returns to privileged EXEC mode.

What to Do Next

After the peer policy template is created, the configuration of the peer policy template can be inherited or applied by another peer policy template. For details about peer policy inheritance, see the "Configuring Peer Policy Template Inheritance with the inherit peer-policy Command" section or the "Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command" section.

Configuring Peer Policy Template Inheritance with the inherit peer-policy Command

This task configures peer policy template inheritance using the **inherit peer-policy**command. It creates and configure a peer policy template and allows it to inherit a configuration from another peer policy template.

When BGP neighbors use inherited peer templates, it can be difficult to determine which policies are associated with a specific template. In Cisco IOS Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, and later releases, the **detail** keyword was added to the **show ip bgp template peer-policy** command to display the detailed configuration of local and inherited policies associated with a specific template.

Note The commands in Steps 5 and 6 are optional and could be replaced with any supported BGP policy configuration commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **route-map** *map-name* {**in**| **out**}
6. **inherit peer-policy** *policy-template-name sequence-number*
7. **end**
8. **show ip bgp template peer-policy** [*policy-template-name*[**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-policy** *policy-template-name* **Example:** `Router(config-router)# template peer-policy NETWORK1`	Enter policy-template configuration mode and creates a peer policy template.

	Command or Action	Purpose
Step 5	**route-map** *map-name* {**in**| **out**} **Example:** `Router(config-router-ptmp)# route-map ROUTE in`	(Optional) Applies the specified route map to inbound or outbound routes. **Note** Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the Configuring Peer Policy Templates, on page 631.
Step 6	**inherit peer-policy** *policy-template-name sequence-number* **Example:** `Router(config-router-ptmp)# inherit peer-policy GLOBAL 10`	Configures the peer policy template to inherit the configuration of another peer policy template. • The *sequence-number* argument sets the order in which the peer policy template is evaluated. Like a route map sequence number, the lowest sequence number is evaluated first. • The example configures this peer policy template to inherit the configuration from GLOBAL. If the template created in these steps is applied to a neighbor, the configuration GLOBAL will also be inherited and applied indirectly. Up to six additional peer policy templates can be indirectly inherited from GLOBAL for a total of eight directly applied and indirectly inherited peer policy templates. • This template in the example will be evaluated first if no other templates are configured with a lower sequence number.
Step 7	**end** **Example:** `Router(config-router-ptmp)# end`	Exits policy-template configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp template peer-policy** [*policy-template-name*[**detail**]] **Example:** `Router# show ip bgp template peer-policy NETWORK1 detail`	Displays locally configured peer policy templates. • The output can be filtered to display a single peer policy template with the *policy-template-name* argument. This command also supports all standard output modifiers. • Use the **detail** keyword to display detailed policy information. **Note** The **detail** keyword is supported only in Cisco IOS Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, and later releases.

Examples

The following sample output of the **show ip bgp template peer-policy** command with the **detail** keyword displays details of the policy named NETWORK1. The output in this example shows that the GLOBAL template was inherited. Details of route map and prefix list configurations are also displayed.

```
Router# show ip bgp template peer-policy NETWORK1 detail
Template:NETWORK1, index:2.
Local policies:0x1, Inherited polices:0x80840
This template inherits:
  GLOBAL, index:1, seq_no:10, flags:0x1
Locally configured policies:
  route-map ROUTE in
Inherited policies:
  prefix-list NO-MARKETING in
  weight 300
  maximum-prefix 10000
 Template:NETWORK1 <detail>
Locally configured policies:
  route-map ROUTE in
route-map ROUTE, permit, sequence 10
  Match clauses:
    ip address prefix-lists: DEFAULT
ip prefix-list DEFAULT: 1 entries
   seq 5 permit 10.1.1.0/24
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
Inherited policies:
  prefix-list NO-MARKETING in
ip prefix-list NO-MARKETING: 1 entries
   seq 5 deny 10.2.2.0/24
```

Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command

This task configures a router to send a peer policy template to a neighbor to inherit using the **neighbor inherit peer-policy** command. Perform the following steps to send a peer policy template configuration to a neighbor to inherit.

When BGP neighbors use multiple levels of peer templates, it can be difficult to determine which policies are applied to the neighbor. In Cisco IOS Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, and later releases, the **policy** and **detail** keywords were added to the **show ip bgp neighbors** command to display the inherited policies and policies configured directly on the specified neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
7. **end**
8. **show ip bgp neighbors** [*ip-address*[**policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	Configures a peering session with the specified neighbor. • The explicit **remote-as** statement is required for the **neighbor inherit** statement in Step 6 to work. If a peering is not configured, the specified neighbor in Step 6 will not accept the session template.
Step 5	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure a neighbor to accept address family-specific command configurations.
Step 6	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* **Example:** `Router(config-router-af)# neighbor 192.168.1.2 inherit peer-policy GLOBAL`	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration. • The example configures a router to send the peer policy template named GLOBAL to the 192.168.1.2 neighbor to inherit. This template can be applied to a neighbor, and if another peer policy template is indirectly inherited from GLOBAL, the indirectly inherited configuration will also be applied. Up to seven additional peer policy templates can be indirectly inherited from GLOBAL.
Step 7	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp neighbors** [*ip-address*[**policy** [**detail**]]] **Example:**	Displays locally configured peer policy templates.

	Command or Action	Purpose
	`Router# show ip bgp neighbors 192.168.1.2 policy`	• The output can be filtered to display a single peer policy template with the *policy-template-name* argument. This command also supports all standard output modifiers. • Use the **policy** keyword to display the policies applied to this neighbor per address family. • Use the **detail** keyword to display detailed policy information. • The **policy** and **detail** keywords are supported only in Cisco IOS Release 12.0(25)S, 12.4(11)T, 12.2(33)SRB, 12.2(33)SB, and later releases. **Note** Only the syntax required for this task is shown. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output shows the policies applied to the neighbor at 192.168.1.2. The output displays both inherited policies and policies configured on the neighbor device. Inherited polices are policies that the neighbor inherits from a peer-group or a peer-policy template.

```
Router# show ip bgp neighbors 192.168.1.2 policy
Neighbor: 192.168.1.2, Address-Family: IPv4 Unicast
Locally configured policies:
 route-map ROUTE in
Inherited polices:
 prefix-list NO-MARKETING in
 route-map ROUTE in
 weight 300
 maximum-prefix 10000
```

Monitoring and Maintaining BGP Dynamic Update Groups

Use this task to clear and display information about the processing of dynamic BGP update groups. The performance of BGP update message generation is improved with the use of BGP update groups. With the configuration of the BGP peer templates and the support of the dynamic BGP update groups, the network operator no longer needs to configure peer groups in BGP and can benefit from improved configuration flexibility and system performance. For information about using BGP peer templates, see the “Configuring Peer Session Templates” and “Configuring Peer Policy Templates” sections.

SUMMARY STEPS

1. **enable**
2. **clear ip bgp update-group** [*index-group* | *ip-address*]
3. **show ip bgp replication** [*index-group* | *ip-address*]

4. **show ip bgp update-group** [*index-group* | *ip-address*] [**summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp update-group** [*index-group*	*ip-address*] **Example:** `Device# clear ip bgp update-group 192.168.2.2`	Clears BGP update group membership and recalculate BGP update groups. • In the example provided, the membership of neighbor 192.168.2.2 is cleared from an update group.
Step 3	**show ip bgp replication** [*index-group*	*ip-address*] **Example:** `Device# show ip bgp replication`	Displays update replication statistics for BGP update groups.
Step 4	**show ip bgp update-group** [*index-group*	*ip-address*] [**summary**] **Example:** `Device# show ip bgp update-group`	Displays information about BGP update groups.

Troubleshooting Tips

Use the **debug ip bgp groups** command to display information about the processing of BGP update groups. Information can be displayed for all update groups, an individual update group, or a specific BGP neighbor. The output of this command can be very verbose. This command should not be deployed in a production network unless your are troubleshooting a problem.

Configuration Examples for a Basic BGP Network

Example: Configuring a BGP Process and Customizing Peers

The following example shows the configuration for Router B in the above (in the "Customizing a BGP Peer" section) with a BGP process configured with two neighbor peers (at Router A and at Router E) in separate autonomous systems. IPv4 unicast routes are exchanged with both peers and IPv4 multicast routes are exchanged with the BGP peer at Router E.

Router B

```
router bgp 45000
 bgp router-id 172.17.1.99
```

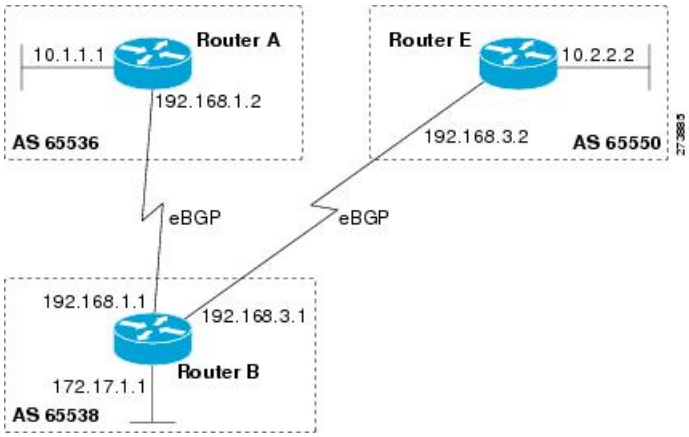
```
no bgp default ipv4-unicast
bgp log-neighbor-changes
timers bgp 70 120
neighbor 192.168.1.2 remote-as 40000
neighbor 192.168.3.2 remote-as 50000
neighbor 192.168.3.2 description finance
!
address-family ipv4
 neighbor 192.168.1.2 activate
 neighbor 192.168.3.2 activate
 no auto-summary
 no synchronization
 network 172.17.1.0 mask 255.255.255.0
 exit-address-family
!
address-family ipv4 multicast
 neighbor 192.168.3.2 activate
 neighbor 192.168.3.2 advertisement-interval 25
 no auto-summary
 no synchronization
 network 172.17.1.0 mask 255.255.255.0
 exit-address-family
```

Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Asplain Format

The following example shows the configuration for Router A, Router B, and Router E in the figure below with a Border Gateway Protocol (BGP) process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using asplain notation. IPv4 unicast routes are exchanged with all peers.

Figure 45: BGP Peers Using 4-Byte Autonomous System Numbers in Asplain Format



Router A

```
router bgp 65536
 bgp router-id 10.1.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
```

```
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.1 remote-as 65538
 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  no auto-summary
  no synchronization
  network 10.1.1.0 mask 255.255.255.0
  exit-address-family
```

Router B

```
router bgp 65538
 bgp router-id 172.17.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 65536
 neighbor 192.168.3.2 remote-as 65550
 neighbor 192.168.3.2 description finance
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

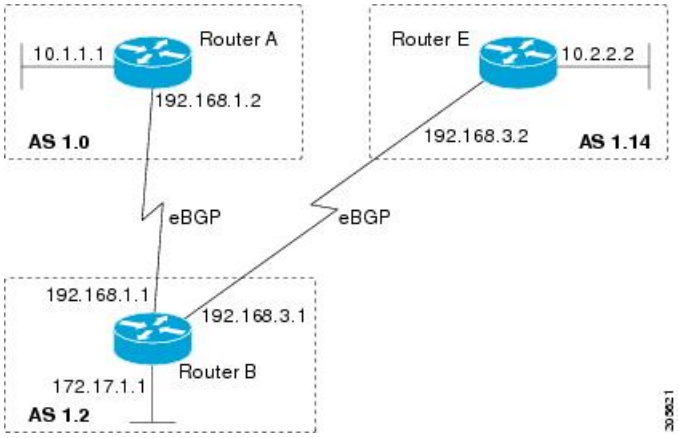
Router E

```
router bgp 65550
 bgp router-id 10.2.2.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.3.1 remote-as 65538
 !
 address-family ipv4
  neighbor 192.168.3.1 activate
  no auto-summary
  no synchronization
  network 10.2.2.0 mask 255.255.255.0
  exit-address-family
```

Asdot Format

The following example shows how to create the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using the default asdot format. IPv4 unicast routes are exchanged with all peers.

Figure 46: BGP Peers Using 4-Byte Autonomous System Numbers in Asdot Format



Router A

```
router bgp 1.0
 bgp router-id 10.1.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.1 remote-as 1.2
 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  no auto-summary
  no synchronization
  network 10.1.1.0 mask 255.255.255.0
  exit-address-family
```

Router B

```
router bgp 1.2
 bgp router-id 172.17.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 1.0
 neighbor 192.168.3.2 remote-as 1.14
 neighbor 192.168.3.2 description finance
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

Router E

```
router bgp 1.14
```

```
 bgp router-id 10.2.2.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.3.1 remote-as 1.2
 !
 address-family ipv4
  neighbor 192.168.3.1 activate
  no auto-summary
  no synchronization
  network 10.2.2.0 mask 255.255.255.0
  exit-address-family
```

Examples: Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXI1, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases and shows how to create a VRF with a route target that uses a 4-byte autonomous system number, 65537, and how to set the route target to extended community value 65537:100 for routes that are permitted by the route map:

```
ip vrf vpn_red
 rd 64500:100
 route-target both 65537:100
 exit
route-map red_map permit 10
 set extcommunity rt 65537:100
 end
```

After the configuration is completed, use the **show route-map** command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 65537:

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
    extended community RT:65537:100
  Policy routing matches: 0 packets, 0 bytes
```

4-Byte Autonomous System Number RD Support

The following example shows how to create a VRF with a route distinguisher that contains a 4-byte AS number 65536, and a route target that contains a 4-byte autonomous system number, 65537:

```
ip vrf vpn_red
 rd 65536:100
 route-target both 65537:100
 exit
```

After the configuration is completed, use the **show vrf** command to verify that the 4-byte AS number route distinguisher is set to 65536:100:

```
RouterB# show vrf vpn_red
Current configuration : 36 bytes
vrf definition x
 rd 65536:100
!
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example is available in Cisco IOS Release 12.0(32)S12, and 12.4(24)T and shows how to create a VRF with a route target that uses a 4-byte autonomous system number, 1.1, and how to set the route target to the extended community value 1.1:100 for routes that are permitted by the route map.

Note In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SXI1, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

```
ip vrf vpn_red
 rd 64500:100
 route-target both 1.1:100
 exit
route-map red_map permit 10
 set extcommunity rt 1.1:100
 end
```

After the configuration is completed, use the **show route-map** command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 1.1.

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
    extended community RT:1.1:100
  Policy routing matches: 0 packets, 0 bytes
```

Asdot Default Format for 4-Byte Autonomous System Number RD Support

The following example works if you have configured asdot as the default display format using the **bgp asnotation dot** command:

```
ip vrf vpn_red
 rd 1.0:100
 route-target both 1.1:100
 exit
```

Example: NLRI to AFI Configuration

The following example upgrades an existing router configuration file in the NLRI format to the AFI format and set the router CLI to use only commands in the AFI format:

```
router bgp 60000
 bgp upgrade-cli
```

The **show running-config** command can be used in privileged EXEC mode to verify that an existing router configuration file has been upgraded from the NLRI format to the AFI format. The following sections provide

sample output from a router configuration file in the NLRI format, and the same router configuration file after it has been upgraded to the AFI format with the **bgp upgrade-cli** command in router configuration mode.

Note After a router has been upgraded from the AFI format to the NLRI format with the **bgp upgrade-cli**command, NLRI commands will no longer be accessible or configurable.

Router Configuration File in NLRI Format Before Upgrading

The following sample output is from the **show running-config** command in privileged EXEC mode. The sample output shows a router configuration file, in the NLRI format, prior to upgrading to the AFI format with the **bgp upgrade-cli** command. The sample output is filtered to show only the affected portion of the router configuration.

```
Router# show running-config | begin bgp

router bgp 101
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 505 nlri unicast multicast
 no auto-summary
!
ip default-gateway 10.4.9.1
ip classless
!
!
route-map REDISTRIBUTE-MULTICAST permit 10
 match ip address prefix-list MULTICAST-PREFIXES
 set nlri multicast
!
route-map MULTICAST-PREFIXES permit 10
!
route-map REDISTRIBUTE-UNICAST permit 20
 match ip address prefix-list UNICAST-PREFIXES
 set nlri unicast
!
!
!
line con 0
line aux 0
line vty 0 4
 password PASSWORD
 login
!
end
```

Router Configuration File in AFI Format After Upgrading

The following sample output shows the router configuration file after it has been upgraded to the AFI format. The sample output is filtered to show only the affected portion of the router configuration file.

```
Router# show running-config | begin bgp

router bgp 101
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 505
 no auto-summary
```

```
 !
 address-family ipv4 multicast
  neighbor 10.1.1.1 activate
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family ipv4
  neighbor 10.1.1.1 activate
  no auto-summary
  no synchronization
  exit-address-family
!
ip default-gateway 10.4.9.1
ip classless
!
!
route-map REDISTRIBUTE-MULTICAST_mcast permit 10
 match ip address prefix-list MULTICAST-PREFIXES
!
route-map REDISTRIBUTE-MULTICAST permit 10
 match ip address prefix-list MULTICAST-PREFIXES
!
route-map MULTICAST-PREFIXES permit 10
!
route-map REDISTRIBUTE-UNICAST permit 20
 match ip address prefix-list UNICAST-PREFIXES
!
!
!
line con 0
line aux 0
line vty 0 4
 password PASSWORD
 login
!
end
```

Examples: Removing BGP Configuration Commands Using a Redistribution Example

The following examples show first the CLI configuration to enable the redistribution of BGP routes into EIGRP using a route map and then the CLI configuration to remove the redistribution and route map. Some BGP configuration commands can affect other CLI commands and this example demonstrates how the removal of one command affects another command.

In the first configuration example, a route map is configured to match and set autonomous system numbers. BGP neighbors in three different autonomous systems are configured and activated. An EIGRP routing process is started, and the redistribution of BGP routes into EIGRP using the route map is configured.

CLI to Enable BGP Route Redistribution Into EIGRP

```
route-map bgp-to-eigrp permit 10
 match tag 50000
 set tag 65000
 exit
router bgp 45000
 bgp log-neighbor-changes
 address-family ipv4
```

```
  neighbor 172.16.1.2 remote-as 45000
  neighbor 172.21.1.2 remote-as 45000
  neighbor 192.168.1.2 remote-as 40000
  neighbor 192.168.3.2 remote-as 50000
  neighbor 172.16.1.2 activate
  neighbor 172.21.1.2 activate
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
 exit
router eigrp 100
 redistribute bgp 45000 metric 10000 100 255 1 1500 route-map bgp-to-eigrp
 no auto-summary
 exit
```

In the second configuration example, both the **route-map** command and the **redistribute** command are disabled. If only the route-map command is removed, it does not automatically disable the redistribution. The redistribution will now occur without any matching or filtering. To remove the redistribution configuration, the **redistribute** command must also be disabled.

CLI to Remove BGP Route Redistribution Into EIGRP

```
configure terminal
 no route-map bgp-to-eigrp
 router eigrp 100
  no redistribute bgp 45000
  end
```

Examples: BGP Soft Reset

The following examples show two ways to reset the connection for BGP peer 192.168.1.1.

Example: Dynamic Inbound Soft Reset

The following example shows the command used to initiate a dynamic soft reconfiguration in the BGP peer 192.168.1.1. This command requires that the peer support the route refresh capability.

```
clear ip bgp 192.168.1.1 soft in
```

Example: Inbound Soft Reset Using Stored Information

The following example shows how to enable inbound soft reconfiguration for the neighbor 192.168.1.1. All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is performed later, the stored information will be used to generate a new set of inbound updates.

```
router bgp 100
 neighbor 192.168.1.1 remote-as 200
 neighbor 192.168.1.1 soft-reconfiguration inbound
```

The following example clears the session with the neighbor 192.168.1.1:

```
clear ip bgp 192.168.1.1 soft in
```

Example: Resetting BGP Peers Using 4-Byte Autonomous System Numbers

The following examples show how to clear BGP peers belonging to an autonomous system that uses 4-byte autonomous system numbers. The initial state of the BGP routing table is shown using the **show ip bgp** command, and peers in 4-byte autonomous systems 65536 and 65550 are displayed.

```
RouterB# show ip bgp

BGP table version is 4, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 65536  i
*> 10.2.2.0/24      192.168.3.2              0             0 65550  i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

The **clear ip bgp 65550** command is entered to remove all BGP peers in the 4-byte autonomous system 65550. The ADJCHANGE message shows that the BGP peer at 192.168.3.2 is being reset.

```
RouterB# clear ip bgp 65550
RouterB#
*Nov 30 23:25:27.043: %BGP-5-ADJCHANGE: neighbor 192.168.3.2 Down User reset
```

The **show ip bgp** command is entered again, and only the peer in 4-byte autonomous systems 65536 is now displayed.

```
RouterB# show ip bgp

BGP table version is 5, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 65536  i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

Almost immediately, the next ADJCHANGE message shows that the BGP peer at 192.168.3.2 (in the 4-byte autonomous system 65550) is now back up.

```
RouterB#
*Nov 30 23:25:55.995: %BGP-5-ADJCHANGE: neighbor 192.168.3.2 Up
```

Example: Resetting and Displaying Basic BGP Information

The following example shows how to reset and display basic BGP information.

The **clear ip bgp *** command clears and resets all the BGP neighbor sessions. In Cisco IOS Release 12.2(25)S and later releases, the syntax is **clear ip bgp all**. Specific neighbors or all peers in an autonomous system can be cleared by using the *neighbor-address* and *autonomous-system-number* arguments. If no argument is specified, this command will clear and reset all BGP neighbor sessions.

Note The **clear ip bgp *** command also clears all the internal BGP structures which makes it useful as a troubleshooting tool.

```
Router# clear ip bgp *
```

The **show ip bgp** command is used to display all the entries in the BGP routing table. The following example displays BGP routing table information for the 10.1.1.0 network:

```
Router# show ip bgp 10.1.1.0 255.255.255.0

BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  40000
    192.168.1.2 from 192.168.1.2 (10.1.1.99)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

The **show ip bgp neighbors** command is used to display information about the TCP and BGP connections to neighbors. The following example displays the routes that were advertised from Router B in the figure above (in the "Configuring a BGP Peer for the IPv4 VRF Address Family" section) to its BGP neighbor 192.168.3.2 on Router E:

```
Router# show ip bgp neighbors 192.168.3.2 advertised-routes

BGP table version is 3, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 40000 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
Total number of prefixes 2
```

The **show ip bgp paths**command is used to display all the BGP paths in the database. The following example displays BGP path information for Router B in the figure above (in the "Customizing a BGP Peer" section):

```
Router# show ip bgp paths

Address    Hash Refcount Metric Path
0x2FB5DB0     0        5      0 i
0x2FB5C90     1        4      0 i
0x2FB5C00  1361        2      0 50000 i
0x2FB5D20  2625        2      0 40000 i
```

The **show ip bgp summary**command is used to display the status of all BGP connections. The following example displays BGP routing table information for Router B in the figure above (in the "Customizing a BGP Peer" section:

```
Router# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 45000
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
4/2 BGP path/bestpath attribute entries using 496 bytes of memory
```

```
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 882 total bytes of memory
BGP activity 14/10 prefixes, 16/12 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.1.2     4 40000     667     672        3    0    0 00:03:49        1
192.168.3.2     4 50000     468     467        0    0    0 00:03:49 (NoNeg)
```

Examples: Aggregating Prefixes Using BGP

The following examples show how you can use aggregate routes in BGP either by redistributing an aggregate route into BGP or by using the BGP conditional aggregation routing feature.

In the following example, the **redistribute static** router configuration command is used to redistribute aggregate route 10.0.0.0:

```
ip route 10.0.0.0 255.0.0.0 null 0
!
router bgp 100
 redistribute static
```

The following configuration shows how to create an aggregate entry in the BGP routing table when at least one specific route falls into the specified range. The aggregate route will be advertised as coming from your autonomous system and has the atomic aggregate attribute set to show that information might be missing. (By default, atomic aggregate is set unless you use the **as-set** keyword in the **aggregate-address** router configuration command.)

```
router bgp 100
 aggregate-address 10.0.0.0 255.0.0.0
```

The following example shows how to create an aggregate entry using the same rules as in the previous example, but the path advertised for this route will be an AS_SET consisting of all elements contained in all paths that are being summarized:

```
router bgp 100
 aggregate-address 10.0.0.0 255.0.0.0 as-set
```

The following example shows how to create the aggregate route for 10.0.0.0 and also suppress advertisements of more specific routes to all neighbors:

```
router bgp 100
 aggregate-address 10.0.0.0 255.0.0.0 summary-only
```

The following example configures BGP to not advertise inactive routes:

```
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 unicast
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```

The following example configures a maximum route limit in the VRF named RED and configures BGP to not advertise inactive routes through the VRF named RED:

```
Device(config)# ip vrf RED
Device(config-vrf)# rd 50000:10
Device(config-vrf)# maximum routes 1000 10
```

```
Device(config-vrf)# exit
Device(config)# router bgp 50000
Device(config-router)# address-family ipv4 vrf RED
Device(config-router-af)# bgp suppress-inactive
Device(config-router-af)# end
```

Example: Configuring a BGP Peer Group

The following example shows how to use an address family to configure a peer group so that all members of the peer group are both unicast- and multicast-capable:

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 address-family ipv4 unicast
  neighbor mygroup peer-group
  neighbor 192.168.1.2 peer-group mygroup
  neighbor 192.168.3.2 peer-group mygroup
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 address-family ipv4 multicast
  neighbor mygroup peer-group
  neighbor 192.168.1.2 peer-group mygroup
  neighbor 192.168.3.2 peer-group mygroup
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
```

Example: Configuring Peer Session Templates

The following example creates a peer session template named INTERNAL-BGP in session-template configuration mode:

```
router bgp 45000
 template peer-session INTERNAL-BGP
 remote-as 50000
 timers 30 300
 exit-peer-session
```

The following example creates a peer session template named CORE1. This example inherits the configuration of the peer session template named INTERNAL-BGP.

```
router bgp 45000
 template peer-session CORE1
 description CORE-123
 update-source loopback 1
 inherit peer-session INTERNAL-BGP
 exit-peer-session
```

The following example configures the 192.168.3.2 neighbor to inherit the CORE1 peer session template. The 192.168.3.2 neighbor will also indirectly inherit the configuration from the peer session template named INTERNAL-BGP. The explicit **remote-as** statement is required for the neighbor inherit statement to work. If a peering is not configured, the specified neighbor will not accept the session template.

```
router bgp 45000
```

```
 neighbor 192.168.3.2 remote-as 50000
 neighbor 192.168.3.2 inherit peer-session CORE1
```

Examples: Configuring Peer Policy Templates

The following example creates a peer policy template named GLOBAL and enters policy-template configuration mode:

```
router bgp 45000
 template peer-policy GLOBAL
  weight 1000
  maximum-prefix 5000
  prefix-list NO_SALES in
  exit-peer-policy
```

The following example creates a peer policy template named PRIMARY-IN and enters policy-template configuration mode:

```
router bgp 45000
 template peer-policy PRIMARY-IN
  prefix-list ALLOW-PRIMARY-A in
  route-map SET-LOCAL in
  weight 2345
  default-originate
  exit-peer-policy
```

The following example creates a peer policy template named CUSTOMER-A. This peer policy template is configured to inherit the configuration from the peer policy templates named PRIMARY-IN and GLOBAL.

```
router bgp 45000
 template peer-policy CUSTOMER-A
  route-map SET-COMMUNITY in
  filter-list 20 in
  inherit peer-policy PRIMARY-IN 20
  inherit peer-policy GLOBAL 10
  exit-peer-policy
```

The following example configures the 192.168.2.2 neighbor in address family mode to inherit the peer policy template named CUSTOMER-A. Assuming this example is a continuation of the example above, because the peer policy template named CUSTOMER-A above inherited the configuration from the templates named PRIMARY-IN and GLOBAL, the 192.168.2.2 neighbor will also indirectly inherit the peer policy templates named PRIMARY-IN and GLOBAL.

```
router bgp 45000
 neighbor 192.168.2.2 remote-as 50000
 address-family ipv4 unicast
  neighbor 192.168.2.2 inherit peer-policy CUSTOMER-A
  end
```

Examples: Monitoring and Maintaining BGP Dynamic Update Peer-Groups

No configuration is required to enable the BGP dynamic update of peer groups and the algorithm runs automatically. The following examples show how BGP update group information can be cleared or displayed.

clear ip bgp update-group Example

The following example clears the membership of neighbor 10.0.0.1 from an update group:

```
Router# clear ip bgp update-group 10.0.0.1
```

debug ip bgp groups Example

The following example output from the **debug ip bgp groups** command shows the recalculation of update groups after the **clear ip bgp groups** command was issued:

```
Router# debug ip bgp groups

5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.5 flags 0x0 cap 0x0 and updgrp 2 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.5 fl0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.8 flags 0x0 cap 0x0 and updgrp 2 fl0
5w4d: BGP-DYN(0): Update-group 2 flags 0x0 cap 0x0 policies same as 10.4.9.8 fl0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Down User reset
5w4d: BGP-DYN(0): Comparing neighbor 10.4.9.21 flags 0x0 cap 0x0 and updgrp 1 f0
5w4d: BGP-DYN(0): Update-group 1 flags 0x0 cap 0x0 policies same as 10.4.9.21 f0
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.5 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.21 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 10.4.9.8 Up
```

show ip bgp replication Example

The following sample output from the **show ip bgp replication** command shows update group replication information for all for neighbors:

```
Router# show ip bgp replication

BGP Total Messages Formatted/Enqueued : 0/0
     Index     Type  Members          Leader   MsgFmt  MsgRepl  Csize  Qsize
         1 internal        1       10.4.9.21        0        0      0      0
         2 internal        2        10.4.9.5        0        0      0      0
```

show ip bgp update-group Example

The following sample output from the **show ip bgp update-group** command shows update group information for all neighbors:

```
Router# show ip bgp update-group

BGP version 4 update-group 1, internal, Address Family: IPv4 Unicast
  BGP Update version : 0, messages 0/0
  Route map for outgoing advertisements is COST1
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
  Minimum time between advertisement runs is 5 seconds
  Has 1 member:
  10.4.9.21
BGP version 4 update-group 2, internal, Address Family: IPv4 Unicast
  BGP Update version : 0, messages 0/0
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
  Minimum time between advertisement runs is 5 seconds
```

```
Has 2 members:
10.4.9.5 10.4.9.8
```

Where to Go Next

- If you want to connect to an external service provider, see the "Connecting to a Service Provider Using External BGP" module.
- To configure BGP neighbor session options, proceed to the "Configuring BGP Neighbor Session Options" module.
- If you want to configure some iBGP features, see the "Configuring Internal BGP Features" module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	"Cisco BGP Overview" module in the *IP Routing: BGP Configuration Guide*
Multiprotocol Label Switching (MPLS) and BGP configuration example using the IPv4 VRF address family	"MPLS VPN Inter-AS with ASBRs Exchanging IPv4 Routes and MPLS Labels" module in the *MPLS: Layer 3 VPNs: Inter-AS and CSC Configuration Guide*

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2519	*A Framework for Inter-Domain Route Aggregation*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 4271	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4893	*BGP Support for Four-octet AS Number Space*
RFC 5396	*Textual Representation of Autonomous system (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

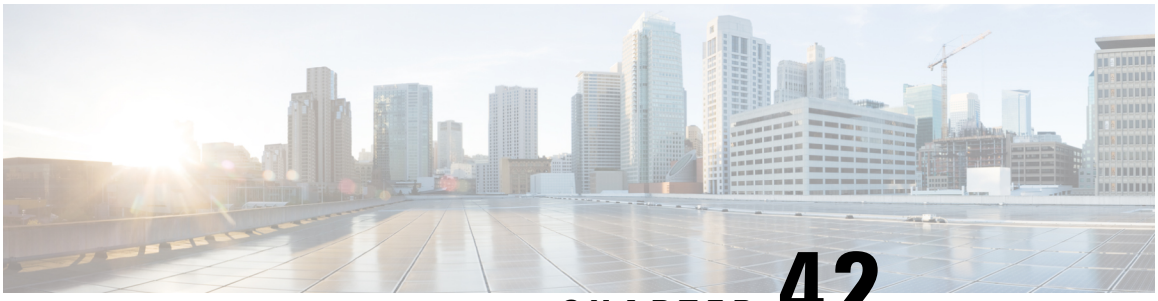
Feature Information for Configuring a Basic BGP Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 60: Feature Information for Configuring a Basic BGP Network

Feature Name	Releases	Feature Configuration Information
BGP Conditional Route Injection	12.0(22)S 12.2(4)T 12.2(14)S 15.0(1)S Cisco IOS XE 3.1.0SG	The BGP Conditional Route Injection feature allows you to inject more specific prefixes into a BGP routing table over less specific prefixes that were selected through normal route aggregation. These more specific prefixes can be used to provide a finer granularity of traffic engineering or administrative control than is possible with aggregated routes.
BGP Configuration Using Peer Templates	12.0(24)S 12.2(18)S 12.2(27)SBC 12.3(4)T 15.0(1)S	The BGP Configuration Using Peer Templates feature introduces a new mechanism that groups distinct neighbor configurations for BGP neighbors that share policies. This type of policy configuration has been traditionally configured with BGP peer groups. However, peer groups have certain limitations because peer group configuration is bound to update grouping and specific session characteristics. Configuration templates provide an alternative to peer group configuration and overcome some of the limitations of peer groups.
BGP Dynamic Update Peer Groups	12.0(24)S 12.2(18)S 12.2(27)SBC 12.3(4)T 15.0(1)S Cisco IOS XE 3.1.0SG	The BGP Dynamic Update Peer Groups feature introduces a new algorithm that dynamically calculates and optimizes update groups of neighbors that share the same outbound policies and can share the same update messages. In previous versions of Cisco IOS software, BGP update messages were grouped based on peer-group configurations. This method of grouping updates limited outbound policies and specific-session configurations. The BGP Dynamic Update Peer Group feature separates update group replication from peer group configuration, which improves convergence time and flexibility of neighbor configuration.
BGP Hybrid CLI	12.0(22)S 12.2(15)T 15.0(1)S	The BGP Hybrid CLI feature simplifies the migration of BGP networks and existing configurations from the NLRI format to the AFI format. This new functionality allows the network operator to configure commands in the AFI format and save these command configurations to existing NLRI formatted configurations. The feature provides the network operator with the capability to take advantage of new features and provides support for migration from the NLRI format to the AFI format.
Suppress BGP Advertisement for Inactive Routes	12.2(25)S 12.2(33)SXH 15.0(1)M 15.0(1)S	The Suppress BGP Advertisements for Inactive Routes feature allows you to configure the suppression of advertisements for routes that are not installed in the Routing Information Base (RIB). Configuring this feature allows Border Gateway Protocol (BGP) updates to be more consistent with data used for traffic forwarding.



CHAPTER 42

BGP 4 Soft Configuration

BGP4 soft configuration allows BGP4 policies to be configured and activated without clearing the BGP session, hence without invalidating the forwarding cache.

Information About BGP 4 Soft Configuration

BGP Session Reset

Whenever the routing policy changes due to a configuration change, BGP peering sessions must be reset by using the **clear ip bgp** command. Cisco software supports the following three mechanisms to reset BGP peering sessions:

- Hard reset—A hard reset tears down the specified peering sessions including the TCP connection and deletes routes coming from the specified peer.
- Soft reset—A soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions. Soft reconfiguration uses stored update information, at the cost of additional memory for storing the updates, to allow you to apply new BGP policy without disrupting the network. Soft reconfiguration can be configured for inbound or outbound sessions.
- Dynamic inbound soft reset—The route refresh capability, as defined in RFC 2918, allows the local device to reset inbound routing tables dynamically by exchanging route refresh requests to supporting peers. The route refresh capability does not store update information locally for nondisruptive policy changes. It instead relies on dynamic exchange with supporting peers. Route refresh must first be advertised through BGP capability negotiation between peers. All BGP devices must support the route refresh capability. To determine if a BGP device supports this capability, use the **show ip bgp neighbors** command. The following message is displayed in the output when the device supports the route refresh capability:

```
Received route refresh capability from peer.
```

The **bgp soft-reconfig-backup** command was introduced to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command.

How to Configure BGP 4 Soft Configuration

Configuring Inbound Soft Reconfiguration When Route Refresh Capability Is Missing

Perform this task to configure inbound soft reconfiguration using the **bgp soft-reconfig-backup** command for BGP peers that do not support the route refresh capability. BGP peers that support the route refresh capability are unaffected by the configuration of this command. Note that the memory requirements for storing the inbound update information can become quite large.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp log-neighbor-changes**
5. **bgp soft-reconfig-backup**
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [**inbound**]
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.
10. **exit**
11. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address*
13. **end**
14. **show ip bgp neighbors** [*neighbor-address*]
15. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**bgp soft-reconfig-backup** **Example:** `Device(config-router)# bgp soft-reconfig-backup`	Configures a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability. • This command is used to configure BGP to perform inbound soft reconfiguration for peers that do not support the route refresh capability. The configuration of this command allows you to configure BGP to store updates (soft reconfiguration) only as necessary. Peers that support the route refresh capability are unaffected by the configuration of this command.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration** [**inbound**] **Example:** `Device(config-router)# neighbor 192.168.1.2 soft-reconfiguration inbound`	Configures the Cisco software to start storing updates. • All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is done later, the stored information will be used to generate a new set of inbound updates.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router)# neighbor 192.168.1.2 route-map LOCAL in`	Applies a route map to incoming or outgoing routes. • In this example, the route map named LOCAL will be applied to incoming routes.
Step 9	Repeat Steps 6 through 8 for every peer that is to be configured with inbound soft reconfiguration.	—
Step 10	**exit** **Example:**	Exits router configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-router)# exit`	
Step 11	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map LOCAL permit 10`	Configures a route map and enters route-map configuration mode. • In this example, a route map named LOCAL is created.
Step 12	**set ip next-hop** *ip-address* **Example:** `Device(config-route-map)# set ip next-hop 192.168.1.144`	Specifies where output packets that pass a match clause of a route map for policy routing. • In this example, the ip address is set to 192.168.1.144.
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	**show ip bgp neighbors** [*neighbor-address*] **Example:** `Device# show ip bgp neighbors 192.168.1.2`	(Optional) Displays information about the TCP and BGP connections to neighbors. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 15	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following partial output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.2.1. This peer supports route refresh.

```
BGP neighbor is 192.168.1.2,  remote AS 40000, external link
 Neighbor capabilities:
   Route refresh: advertised and received(new)
```

The following partial output from the **show ip bgp neighbors** command shows information about the TCP and BGP connections to the BGP neighbor 192.168.3.2. This peer does not support route refresh so the soft-reconfig inbound paths for BGP peer 192.168.3.2 will be stored because there is no other way to update any inbound policy updates.

```
BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Neighbor capabilities:
   Route refresh: advertised
```

The following sample output from the **show ip bgp** command shows the entry for the network 172.17.1.0. Both BGP peers are advertising 172.17.1.0/24, but only the received-only path is stored for 192.168.3.2.

```
BGP routing table entry for 172.17.1.0/24, version 11
Paths: (3 available, best #3, table Default-IP-Routing-Table, RIB-failure(4))
Flag: 0x820
 Advertised to update-groups:
    1
 50000
   192.168.3.2 from 192.168.3.2 (172.17.1.0)
     Origin incomplete, metric 0, localpref 200, valid, external
 50000, (received-only)
   192.168.3.2 from 192.168.3.2 (172.17.1.0)
     Origin incomplete, metric 0, localpref 100, valid, external
 40000
   192.168.1.2 from 192.168.1.2 (172.16.1.0)
     Origin incomplete, metric 0, localpref 200, valid, external, best
```

Configuration Examples for BGP 4 Soft Configuration

Examples: BGP Soft Reset

The following examples show two ways to reset the connection for BGP peer 192.168.1.1.

Example: Dynamic Inbound Soft Reset

The following example shows the command used to initiate a dynamic soft reconfiguration in the BGP peer 192.168.1.1. This command requires that the peer support the route refresh capability.

```
clear ip bgp 192.168.1.1 soft in
```

Example: Inbound Soft Reset Using Stored Information

The following example shows how to enable inbound soft reconfiguration for the neighbor 192.168.1.1. All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is performed later, the stored information will be used to generate a new set of inbound updates.

```
router bgp 100
 neighbor 192.168.1.1 remote-as 200
 neighbor 192.168.1.1 soft-reconfiguration inbound
```

The following example clears the session with the neighbor 192.168.1.1:

```
clear ip bgp 192.168.1.1 soft in
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

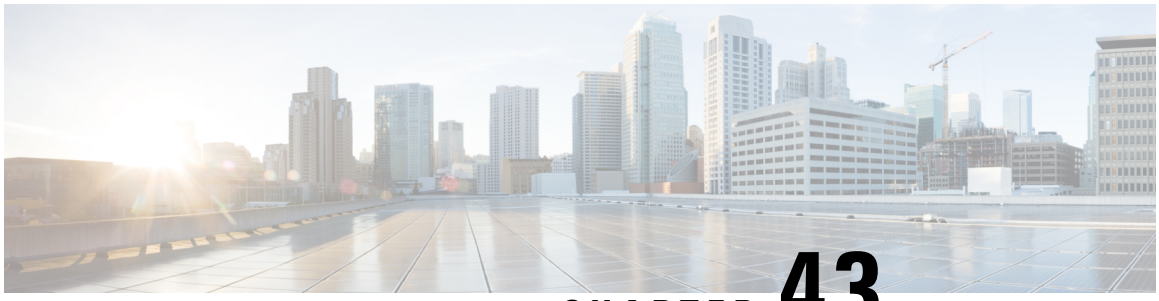
Feature Information for BGP 4 Soft Configuration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 61: Feature Information for BGP 4 Soft Configuration

Feature Name	Releases	Feature Information
BGP 4 Soft Configuration		BGP 4 Soft Configuration allows BGP4 policies to be configured and activated without clearing the BGP session, hence without invalidating the forwarding cache.



CHAPTER 43

BGP Support for 4-byte ASN

The Cisco implementation of 4-byte autonomous system (AS) numbers uses asplain (65538, for example) as the default regular expression match and the output display format for AS numbers. However, you can configure 4-byte AS numbers in both the asplain format and the asdot format as described in RFC 5396. In addition, 4-byte ASN route distinguisher (RD) and route target (RT) BGP support for 4-byte autonomous numbers is added.

Information About BGP Support for 4-byte ASN

BGP Autonomous System Number Formats

Prior to January 2009, BGP autonomous system (AS) numbers that were allocated to companies were 2-octet numbers in the range from 1 to 65535 as described in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*. Due to increased demand for AS numbers, the Internet Assigned Number Authority (IANA) started to allocate four-octet AS numbers in the range from 65536 to 4294967295. RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*, documents three methods of representing AS numbers. Cisco has implemented the following two methods:

- Asplain—Decimal value notation where both 2-byte and 4-byte AS numbers are represented by their decimal value. For example, 65526 is a 2-byte AS number and 234567 is a 4-byte AS number.
- Asdot—Autonomous system dot notation where 2-byte AS numbers are represented by their decimal value and 4-byte AS numbers are represented by a dot notation. For example, 65526 is a 2-byte AS number and 1.169031 is a 4-byte AS number (this is dot notation for the 234567 decimal number).

For details about the third method of representing autonomous system numbers, see RFC 5396.

Asdot Only Autonomous System Number Formatting

In Cisco IOS XE Release 2.3, the 4-octet (4-byte) AS numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte AS numbers the asdot

format includes a period, which is a special character in regular expressions. A backslash must be entered before the period (for example, 1\.14) to ensure the regular expression match does not fail. The table below shows the format in which 2-byte and 4-byte AS numbers are configured, matched in regular expressions, and displayed in **show** command output in Cisco IOS images where only asdot formatting is available.

Table 62: Asdot Only 4-Byte AS Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default AS Number Formatting

In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte AS numbers uses asplain as the default display format for AS numbers, but you can configure 4-byte AS numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte AS numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte AS numbers are written in the asplain format. If you want to change the default **show** command output to display 4-byte autonomous system numbers in the asdot format, use the **bgp asnotation dot** command under router configuration mode. When the asdot format is enabled as the default, any regular expressions to match 4-byte AS numbers must be written using the asdot format, or the regular expression match will fail. The tables below show that although you can configure 4-byte AS numbers in either asplain or asdot format, only one format is used to display **show** command output and control 4-byte AS number matching for regular expressions, and the default is asplain format. To display 4-byte AS numbers in **show** command output and to control matching for regular expressions in the asdot format, you must configure the **bgp asnotation dot** command. After enabling the **bgp asnotation dot** command, a hard reset must be initiated for all BGP sessions by entering the **clear ip bgp *** command.

Note If you are upgrading to an image that supports 4-byte AS numbers, you can still use 2-byte AS numbers. The **show** command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte AS numbers regardless of the format configured for 4-byte AS numbers.

Table 63: Default Asplain 4-Byte AS Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 64: Asdot 4-Byte AS Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private AS Numbers

In Cisco IOS XE Release 2.3 and later releases, the Cisco implementation of BGP supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte AS numbers to 4-byte AS numbers. A new reserved (private) AS number, 23456, was created by RFC 4893 and this number cannot be configured as an AS number in the Cisco IOS CLI.

RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use*, describes new reserved AS numbers for documentation purposes. Use of the reserved numbers allow configuration examples to be accurately documented and avoids conflict with production networks if these configurations are literally copied. The reserved numbers are documented in the IANA AS number registry. Reserved 2-byte AS numbers are in the contiguous block, 64496 to 64511 and reserved 4-byte AS numbers are from 65536 to 65551 inclusive.

Private 2-byte AS numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private AS numbers can be used for internal routing domains but must be translated for traffic that is routed out to the Internet. BGP should not be configured to advertise private AS numbers to external networks. Cisco IOS software does not remove private AS numbers from routing updates by default. We recommend that ISPs filter private AS numbers.

Note AS number assignment for public and private networks is governed by the IANA. For information about AS numbers, including reserved number assignment, or to apply to register an AS number, see the following URL: http://www.iana.org/.

Cisco Implementation of 4-Byte Autonomous System Numbers

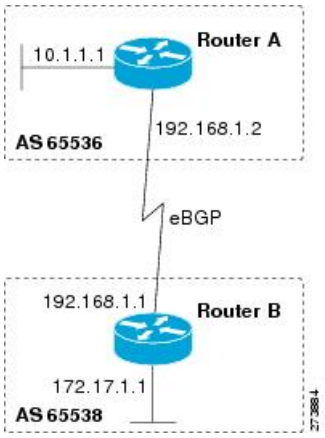
In Cisco IOS XE Release 2.4 and later releases, the Cisco implementation of 4-byte autonomous system (AS) numbers uses asplain—65538, for example—as the default regular expression match and output display format for AS numbers, but you can configure 4-byte AS numbers in both the asplain format and the asdot format as described in RFC 5396. To change the default regular expression match and output display of 4-byte AS numbers to asdot format, use the **bgp asnotation dot** command followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions. For more details about 4-byte AS number formats, see the "BGP Autonomous System Number Formats" section.

In Cisco IOS XE Release 2.3, the Cisco implementation of 4-byte AS numbers uses asdot—1.2, for example—as the only configuration format, regular expression match, and output display, with no asplain support. For an example of BGP peers in two autonomous systems using 4-byte numbers, see the figure below. To view a configuration example of the configuration between three neighbor peers in separate 4-byte autonomous systems configured using asdot notation, see the "Example: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers" section.

Cisco also supports RFC 4893, which was developed to allow BGP to support a gradual transition from 2-byte AS numbers to 4-byte AS numbers. To ensure a smooth transition, we recommend that all BGP speakers within an AS that is identified using a 4-byte AS number be upgraded to support 4-byte AS numbers.

Note A new private AS number, 23456, was created by RFC 4893, and this number cannot be configured as an AS number in the Cisco IOS CLI.

Figure 47: BGP Peers in Two Autonomous Systems Using 4-Byte Numbers



How to Configure BGP Support for 4-byte ASN

Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Perform this task to configure a Border Gateway Protocol (BGP) routing process and BGP peers when the BGP peers are located in an autonomous system (AS) that uses 4-byte AS numbers. The address family configured here is the default IPv4 unicast address family, and the configuration is done at Router B in the figure above (in the "Cisco Implementation of 4-Byte Autonomous System Numbers" section). The 4-byte AS numbers in this task are formatted in the default asplain (decimal value) format; for example, Router B is in AS number 65538 in the figure above. Remember to perform this task for any neighbor routers that are to be BGP peers.

Before you begin

Note By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. Repeat Step 4 to define other BGP neighbors, as required.
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **neighbor** *ip-address* **activate**
8. Repeat Step 7 to activate other BGP neighbors, as required.
9. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
10. **end**
11. **show ip bgp** [*network*] [*network-mask*]
12. **show ip bgp summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 65536`	Adds the IP address of the neighbor in the specified AS to the IPv4 multiprotocol BGP neighbor table of the local device. • In this example, the 4-byte AS number, 65536, is defined in asplain notation.
Step 5	Repeat Step 4 to define other BGP neighbors, as required.	--
Step 6	**address-family ipv4** [**unicast**	**multicast**	**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the device is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes.

	Command or Action	Purpose
		• The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local device.
Step 8	Repeat Step 7 to activate other BGP neighbors, as required.	--
Step 9	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this AS and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp 10.1.1.0`	(Optional) Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 12	**show ip bgp summary** **Example:** `Device# show ip bgp summary`	(Optional) Displays the status of all BGP connections.

Examples

The following output from the **show ip bgp** command at Router B shows the BGP routing table entry for network 10.1.1.0 learned from the BGP neighbor at 192.168.1.2 in Router A in the figure above with its 4-byte AS number of 65536 displayed in the default asplain format.

```
RouterB# show ip bgp 10.1.1.0

BGP routing table entry for 10.1.1.0/24, version 2
Paths: (1 available, best #1)
  Advertised to update-groups:
     2
```

```
  65536
    192.168.1.2 from 192.168.1.2 (10.1.1.99)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

The following output from the **show ip bgp summary** command shows the 4-byte AS number 65536 for the BGP neighbor 192.168.1.2 of Router A in the figure above after this task has been configured on Router B:

```
RouterB# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
3/2 BGP path/bestpath attribute entries using 444 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 806 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  Stated
192.168.1.2     4       65536       6       6        3    0    0 00:01:33       1
```

Troubleshooting Tips

Use the **ping** command to verify basic network connectivity between the BGP devices.

Modifying the Default Output and Regular Expression Match Format for 4-Byte Autonomous System Numbers

Perform this task to modify the default output format for 4-byte autonomous system (AS) numbers from asplain format to asdot notation format. The **show ip bgp summary** command is used to display the changes in output format for the 4-byte AS numbers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp summary**
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **bgp asnotation dot**
6. **end**
7. **clear ip bgp ***
8. **show ip bgp summary**
9. **show ip bgp regexp** *regexp*
10. **configure terminal**
11. **router bgp** *autonomous-system-number*
12. **no bgp asnotation dot**
13. **end**
14. **clear ip bgp ***

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp summary** **Example:** `Device# show ip bgp summary`	Displays the status of all Border Gateway Protocol (BGP) connections.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 5	**bgp asnotation dot** **Example:** `Device(config-router)# bgp asnotation dot`	Changes the default output format of BGP 4-byte AS numbers from asplain (decimal values) to dot notation. **Note** 4-byte AS numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for **show** commands or the matching of regular expressions.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 7	**clear ip bgp *** **Example:** `Device# clear ip bgp *`	Clears and resets all current BGP sessions. • In this example, a hard reset is performed to ensure that the 4-byte AS number format change is reflected in all BGP sessions. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 8	**show ip bgp summary** **Example:**	Displays the status of all BGP connections.

	Command or Action	Purpose
	`Device# show ip bgp summary`	
Step 9	**show ip bgp regexp** *regexp* **Example:** `Device# show ip bgp regexp ^1\.0$`	Displays routes that match the AS path regular expression. • In this example, a regular expression to match a 4-byte AS path is configured using asdot format.
Step 10	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 11	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65538`	Enters router configuration mode for the specified routing process. • In this example, the 4-byte AS number, 65538, is defined in asplain notation.
Step 12	**no bgp asnotation dot** **Example:** `Device(config-router)# no bgp asnotation dot`	Resets the default output format of BGP 4-byte AS numbers back to asplain (decimal values). **Note** 4-byte AS numbers can be configured using either asplain format or asdot format. This command affects only the output displayed for **show** commands or the matching of regular expressions.
Step 13	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 14	**clear ip bgp *** **Example:** `Device# clear ip bgp *`	Clears and resets all current BGP sessions. • In this example, a hard reset is performed to ensure that the 4-byte AS number format change is reflected in all BGP sessions. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following output from the **show ip bgp summary** command shows the default asplain format of the 4-byte AS numbers. Note the asplain format of the 4-byte AS numbers, 65536 and 65550.

```
Router# show ip bgp summary
```

```
BGP router identifier 172.17.1.99, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4        65536       7       7        1    0    0 00:03:04      0
192.168.3.2     4        65550       4       4        1    0    0 00:00:15      0
```

After the **bgp asnotation dot** command is configured (followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions), the output is converted to asdot notation format as shown in the following output from the **show ip bgp summary** command. Note the asdot format of the 4-byte AS numbers, 1.0 and 1.14 (these are the asdot conversions of the 65536 and 65550 AS numbers.

```
Router# show ip bgp summary

BGP router identifier 172.17.1.99, local AS number 1.2
BGP table version is 1, main routing table version 1
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
192.168.1.2     4          1.0       9       9        1    0    0 00:04:13      0
192.168.3.2     4         1.14       6       6        1    0    0 00:01:24      0
```

After the **bgp asnotation dot** command is configured (followed by the **clear ip bgp *** command to perform a hard reset of all current BGP sessions), the regular expression match format for 4-byte AS paths is changed to asdot notation format. Although a 4-byte AS number can be configured in a regular expression using either asplain format or asdot format, only 4-byte AS numbers configured using the current default format are matched. In the first example below, the **show ip bgp regexp** command is configured with a 4-byte AS number in asplain format. The match fails because the default format is currently asdot format and there is no output. In the second example using asdot format, the match passes and the information about the 4-byte AS path is shown using the asdot notation.

Note The asdot notation uses a period, which is a special character in Cisco regular expressions. To remove the special meaning, use a backslash before the period.

```
Router# show ip bgp regexp ^65536$

Router# show ip bgp regexp ^1\.0$

BGP table version is 2, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 1.0 i
```

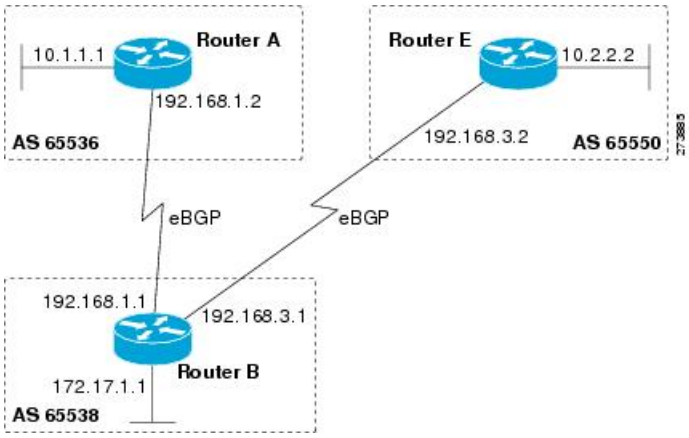
Configuration Examples for BGP Support for 4-byte ASN

Examples: Configuring a BGP Routing Process and Peers Using 4-Byte Autonomous System Numbers

Asplain Format

The following example shows the configuration for Router A, Router B, and Router E in the figure below with a Border Gateway Protocol (BGP) process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using asplain notation. IPv4 unicast routes are exchanged with all peers.

Figure 48: BGP Peers Using 4-Byte Autonomous System Numbers in Asplain Format



Router A

```
router bgp 65536
 bgp router-id 10.1.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.1 remote-as 65538
 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  no auto-summary
  no synchronization
  network 10.1.1.0 mask 255.255.255.0
  exit-address-family
```

Router B

```
router bgp 65538
 bgp router-id 172.17.1.99
 no bgp default ipv4-unicast
```

```
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 65536
 neighbor 192.168.3.2 remote-as 65550
 neighbor 192.168.3.2 description finance
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

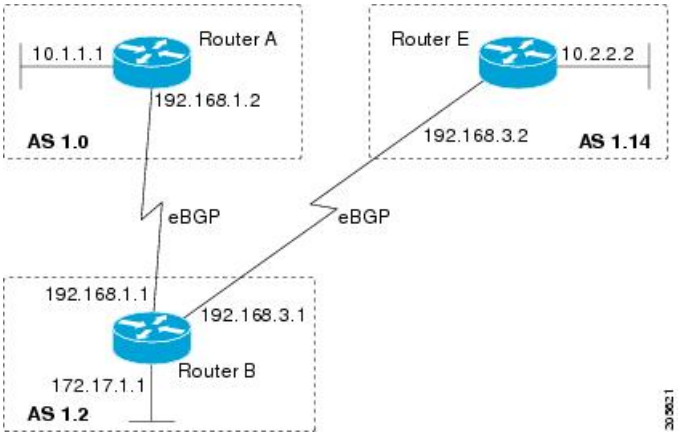
Router E

```
router bgp 65550
 bgp router-id 10.2.2.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.3.1 remote-as 65538
 !
 address-family ipv4
  neighbor 192.168.3.1 activate
  no auto-summary
  no synchronization
  network 10.2.2.0 mask 255.255.255.0
  exit-address-family
```

Asdot Format

The following example shows how to create the configuration for Router A, Router B, and Router E in the figure below with a BGP process configured between three neighbor peers (at Router A, at Router B, and at Router E) in separate 4-byte autonomous systems configured using the default asdot format. IPv4 unicast routes are exchanged with all peers.

Figure 49: BGP Peers Using 4-Byte Autonomous System Numbers in Asdot Format



Router A

```
router bgp 1.0
 bgp router-id 10.1.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.1 remote-as 1.2
 !
 address-family ipv4
  neighbor 192.168.1.1 activate
  no auto-summary
  no synchronization
  network 10.1.1.0 mask 255.255.255.0
  exit-address-family
```

Router B

```
router bgp 1.2
 bgp router-id 172.17.1.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 1.0
 neighbor 192.168.3.2 remote-as 1.14
 neighbor 192.168.3.2 description finance
 !
 address-family ipv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  no auto-summary
  no synchronization
  network 172.17.1.0 mask 255.255.255.0
  exit-address-family
```

Router E

```
router bgp 1.14
 bgp router-id 10.2.2.99
 no bgp default ipv4-unicast
 bgp fast-external-fallover
 bgp log-neighbor-changes
 timers bgp 70 120
 neighbor 192.168.3.1 remote-as 1.2
 !
 address-family ipv4
  neighbor 192.168.3.1 activate
  no auto-summary
  no synchronization
  network 10.2.2.0 mask 255.255.255.0
  exit-address-family
```

Examples: Configuring a VRF and Setting an Extended Community Using a BGP 4-Byte Autonomous System Number

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXI1, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases and shows how to create a VRF with a route target that uses a 4-byte autonomous system number, 65537, and how to set the route target to extended community value 65537:100 for routes that are permitted by the route map:

```
ip vrf vpn_red
 rd 64500:100
 route-target both 65537:100
 exit
route-map red_map permit 10
 set extcommunity rt 65537:100
 end
```

After the configuration is completed, use the **show route-map** command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 65537:

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
    extended community RT:65537:100
  Policy routing matches: 0 packets, 0 bytes
```

4-Byte Autonomous System Number RD Support

The following example shows how to create a VRF with a route distinguisher that contains a 4-byte AS number 65536, and a route target that contains a 4-byte autonomous system number, 65537:

```
ip vrf vpn_red
 rd 65536:100
 route-target both 65537:100
 exit
```

After the configuration is completed, use the **show vrf** command to verify that the 4-byte AS number route distinguisher is set to 65536:100:

```
RouterB# show vrf vpn_red
Current configuration : 36 bytes
vrf definition x
 rd 65536:100
!
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example is available in Cisco IOS Release 12.0(32)S12, and 12.4(24)T and shows how to create a VRF with a route target that uses a 4-byte autonomous system number, 1.1, and how to set the route target to the extended community value 1.1:100 for routes that are permitted by the route map.

Note In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SXI1, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

```
ip vrf vpn_red
 rd 64500:100
 route-target both 1.1:100
 exit
route-map red_map permit 10
 set extcommunity rt 1.1:100
 end
```

After the configuration is completed, use the **show route-map** command to verify that the extended community is set to the route target that contains the 4-byte autonomous system number of 1.1.

```
RouterB# show route-map red_map
route-map red_map, permit, sequence 10
  Match clauses:
  Set clauses:
    extended community RT:1.1:100
  Policy routing matches: 0 packets, 0 bytes
```

Asdot Default Format for 4-Byte Autonomous System Number RD Support

The following example works if you have configured asdot as the default display format using the **bgp asnotation dot** command:

```
ip vrf vpn_red
 rd 1.0:100
 route-target both 1.1:100
 exit
```

Additional References for BGP Support for 4-byte ASN

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4893	*BGP Support for Four-octet AS Number Space*
RFC 5396	*Textual Representation of Autonomous System (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*

Standard/RFC	Title
RFC 5668	*4-Octet AS Specific BGP Extended Community*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

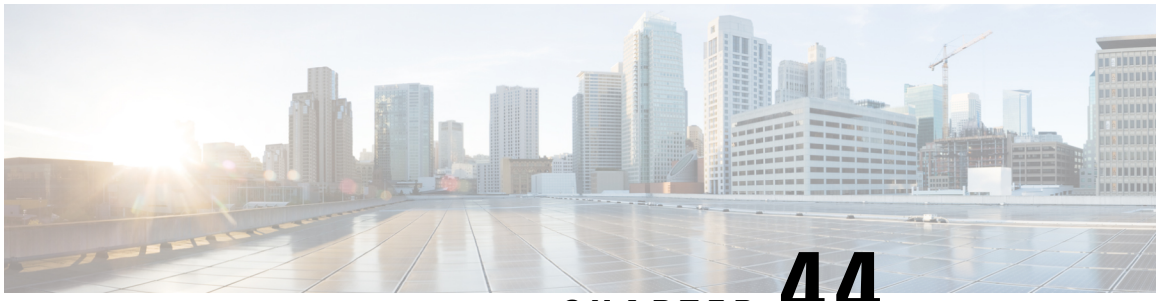
Feature Information for BGP Support for 4-byte ASN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 65: Feature Information for BGP Support for 4-byte ASN

Feature Name	Releases	Feature Information
BGP Support for 4-byte ASN		The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. The following commands were introduced or modified: **bgp asnotation dot**, **bgp confederation identifier**, **bgp confederation peers**, all **clear ip bgp**commands that configure an autonomous system number, **ip as-path access-list**, **ip extcommunity-list**, **match source-protocol**, **neighbor local-as**, **neighbor remote-as**, **redistribute (**IP**)**, **router bgp**, **route-target**, **set as-path**, **set extcommunity**, **set origin**, all **show ip bgp** commands that display an autonomous system number, and **show ip extcommunity-list**.
BGP—4-Byte ASN RD and RT Support		The BGP Support for 4-Byte ASN RD and RT support for 4-byte autonomous system numbers was added.



CHAPTER 44

IPv6 Routing: Multiprotocol BGP Extensions for IPv6

Information About IPv6 Routing: Multiprotocol BGP Extensions for IPv6

Multiprotocol BGP Extensions for IPv6

Multiprotocol BGP is the supported Exterior Gateway Protocol (EGP) for IPv6. Multiprotocol BGP extensions for IPv6 supports many of the same features and functionality as IPv4 BGP. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and Network Layer Reachability Information (NLRI) and next hop (the next device in the path to the destination) attributes that use IPv6 addresses.

How to Implement Multiprotocol BGP for IPv6

Configuring an IPv6 BGP Routing Process and BGP Router ID

Perform this task to configure an IPv6 BGP routing process and an optional BGP router ID for a BGP-speaking device.

BGP uses a router ID to identify BGP-speaking peers. The BGP router ID is 32-bit value that is often represented by an IPv4 address. By default, the router ID is set to the IPv4 address of a loopback interface on the device. If no loopback interface is configured on the device, then the software chooses the highest IPv4 address configured to a physical interface on the device to represent the BGP router ID.

When configuring BGP on a device that is enabled only for IPv6 (that is, the device does not have an IPv4 address), you must manually configure the BGP router ID for the device. The BGP router ID, which is represented as a 32-bit value using an IPv4 address syntax, must be unique to the BGP peers of the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **bgp router-id** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process, and enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** command unless you configure the **no bgp default ipv4-unicast** command before configuring the **neighbor remote-as** command.
Step 5	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id 192.168.99.70`	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. **Note** Configuring a router ID using the **bgp router-id** command resets all active BGP peering sessions.

Configuring IPv6 Multiprotocol BGP Between Two Peers

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes,

neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address* [**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...]
5. **address-family ipv6** [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* [**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] **Example:** `Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 5	**address-family ipv6** [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the **address-family ipv6** command. • The **multicast** keyword specifies IPv6 multicast address prefixes.

	Command or Action	Purpose
Step 6	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate** **Example:** `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate`	Enables the neighbor to exchange prefixes for the IPv6 address family with the local device.

Advertising IPv4 Routes Between IPv6 BGP Peers

If an IPv6 network is connecting two separate IPv4 networks, IPv6 can be used to advertise the IPv4 routes. Configure the peering using the IPv6 addresses within the IPv4 address family. Set the next hop with a static route or with an inbound route map because the advertised next hop will usually be unreachable. Advertising IPv6 routes between two IPv4 peers is also possible using the same model.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** *ipv6-address* **peer-group** *peer-group-name*
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**}
9. **exit**
10. **exit**
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address* [*... ip-address*] [**peer-address**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:**	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 65000`	
Step 4	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router)# neighbor 6peers peer-group`	Creates a multiprotocol BGP peer group.
Step 5	**neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] **Example:** `Device(config-router)# neighbor 6peers remote-as 65002`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local device.
Step 6	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**neighbor** *ipv6-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 2001:DB8:1234::2 peer-group 6peers`	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* [**%**]} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 6peers route-map rmap out`	Applies a route map to incoming or outgoing routes. • Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the **clear bgp ipv6** command with the **soft** and **in** keywords will perform a soft reset.
Step 9	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode, and returns the device to router configuration mode.
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode, and returns the device to global configuration mode.
Step 11	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map rmap permit 10`	Defines a route map and enters route-map configuration mode.

	Command or Action	Purpose
Step 12	**set ip next-hop** *ip-address* [*... ip-address*] [**peer-address**] **Example:** `Device(config-route-map)# set ip next-hop 10.21.8.10`	Overrides the next hop advertised to the peer for IPv4 packets.

Clearing External BGP Peers

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6** {**unicast** | **multicast**} **external** [**soft**] [**in** | **out**]
3. **clear bgp ipv6** {**unicast** | **multicast**} **peer-group** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear bgp ipv6** {**unicast**	**multicast**} **external** [**soft**] [**in**	**out**] **Example:** `Device# clear bgp ipv6 unicast external soft in`	Clears external IPv6 BGP peers.
Step 3	**clear bgp ipv6** {**unicast**	**multicast**} **peer-group** *name* **Example:** `Device# clear bgp ipv6 unicast peer-group marketing`	Clears all members of an IPv6 BGP peer group.

Configuring BGP IPv6 Admin Distance

•

Before you begin

•

SUMMARY STEPS

1.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Example:**	

Example

What to do next

-

Configuration Examples for Multiprotocol BGP for IPv6

Example: Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer

The following example enables IPv6 globally, configures a BGP process, and establishes a BGP router ID. Also, the IPv6 multiprotocol BGP peer 2001:DB8:0:CC00::1 is configured and activated.

```
ipv6 unicast-routing
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp router-id 192.168.99.70
 neighbor 2001:DB8:0:CC00::1 remote-as 64600
 address-family ipv6 unicast
  neighbor 2001:DB8:0:CC00::1 activate
```

Example: Configuring an IPv6 Multiprotocol BGP Peer Group

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 65000
no bgp default ipv4-unicast
neighbor group1 peer-group
neighbor 2001:DB8:0:CC00::1 remote-as 64600
address-family ipv6 unicast
 neighbor group1 activate
 neighbor 2001:DB8:0:CC00::1 peer-group group1
```

Example: Advertising Routes into IPv6 Multiprotocol BGP

The following example injects the IPv6 network 2001:DB8::/24 into the IPv6 unicast database of the local device. (BGP checks that a route for the network exists in the IPv6 unicast database of the local device before advertising the network.)

```
router bgp 65000
```

```
 no bgp default ipv4-unicast
address-family ipv6 unicast
  network 2001:DB8::/24
```

Example: Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes

The following example configures the route map named rtp to permit IPv6 unicast routes from network 2001:DB8::/24 if they match the prefix list named cisco:

```
router bgp 64900
no bgp default ipv4-unicast
neighbor 2001:DB8:0:CC00::1 remote-as 64700
address-family ipv6 unicast
 neighbor 2001:DB8:0:CC00::1 activate
 neighbor 2001:DB8:0:CC00::1 route-map rtp in
ipv6 prefix-list cisco seq 10 permit 2001:DB8::/24
route-map rtp permit 10
 match ipv6 address prefix-list cisco
```

Example: Redistributing Prefixes into IPv6 Multiprotocol BGP

The following example redistributes RIP routes into the IPv6 unicast database of the local device:

```
router bgp 64900
no bgp default ipv4-unicast
address-family ipv6 unicast
 redistribute rip
```

Example: Advertising IPv4 Routes Between IPv6 Peers

The following example advertises IPv4 routes between IPv6 peers when the IPv6 network is connecting two separate IPv4 networks. Peering is configured using IPv6 addresses in the IPv4 address family configuration mode. The inbound route map named rmap sets the next hop because the advertised next hop is likely to be unreachable.

```
router bgp 65000
!
 neighbor 6peers peer-group
 neighbor 2001:DB8:1234::2 remote-as 65002
 address-family ipv4
 neighbor 6peers activate
 neighbor 6peers soft-reconfiguration inbound
 neighbor 2001:DB8:1234::2 peer-group 6peers
 neighbor 2001:DB8:1234::2 route-map rmap in
!
route-map rmap permit 10
 set ip next-hop 10.21.8.10
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

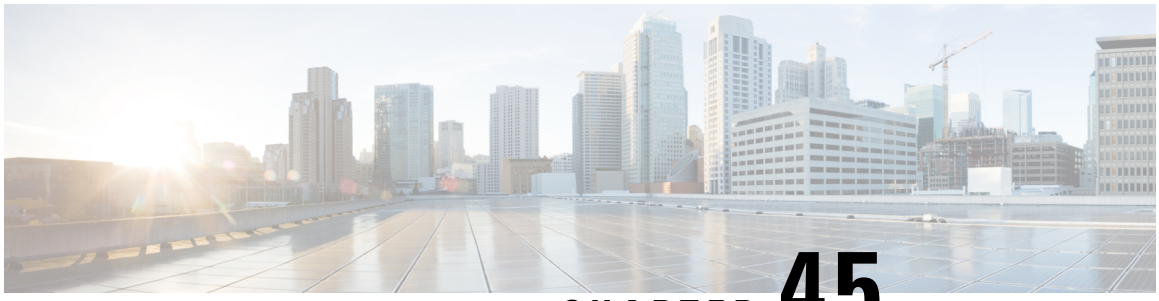
Feature Information for IPv6 Routing Multiprotocol BGP Extensions for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 66: Feature Information for IPv6 Routing Multiprotocol BGP Extensions for IPv6

Feature Name	Releases	Feature Information
IPv6 Routing: Multiprotocol BGP Extensions for IPv6	Cisco IOS XE Release 2.1	Multiprotocol BGP extensions for IPv6 supports the same features and functionality as IPv4 BGP.



CHAPTER 45

IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

Information About IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

IPv6 Multiprotocol BGP Peering Using a Link-Local Address

The IPv6 multiprotocol BGP can be configured between two IPv6 devices (peers) using link-local addresses.

Boarder Gateway Protocol (BGP) uses third-party next hops for peering with multiple peers over IPv6 link-local addresses on the same interface. Peering over link-local addresses on different interfaces cannot use third party next hops. The neighbors peering using link-local addresses are split into one update group per interface. BGP splits update group membership for neighbors with link-local addresses based on the interface used to communicate with that neighbor.

How to Configure IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address

Note

- By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.
- By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.
- The route-map used to modify the next hop needs to be applied outbound only. Inbound route-map to modify next-hop ipv6 address is not supported. Inbound route-map is supported only for IPV4 address family.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ipv6-address* **%** *interface-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number ...*]
5. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%** *interface-name*} **activate**
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*[**%** *interface-name*]} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. Repeat Step 8.
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
12. **set ipv6 next-hop** *ipv6-address* [*link-local-address*] [**peer-address**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ipv6-address* **%** *interface-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number* ...] **Example:** `Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% remote-as 64600`	Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. **Note** Interface for BGP Link-Local neighbor addresses must be configured as part of the address, for example: `FE80::1234:BFF:FE0E:A471%GigabitEthernet0/0/0` This configuration allows you to have the same local link peering address in multiple interfaces.
Step 5	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the **unicast** keyword is not specified with the **address-family ipv6** command. • The **multicast** keyword specifies IPv6 multicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%** *interface-name*} **activate** **Example:** `Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% activate`	Enables the neighbor to exchange prefixes for the IPv6 address family with the local router using the specified link-local addresses. • The optional **%** keyword is the IPv6 link-local address identifier. This keyword needs to be added whenever a link-local IPv6 address is used outside the context of its interface.
Step 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*[**%** *interface-name*]} **route-map** *map-name* {**in** | **out**} **Example:**	Applies a route map to incoming or outgoing routes. • The optional **%** keyword is the IPv6 link-local address identifier. This keyword needs to be added

	Command or Action	Purpose
	`Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% route-map nh6 out`	whenever a link-local IPv6 address is used outside the context of its interface.
Step 8	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode, and returns the device to router configuration mode.
Step 9	Repeat Step 8. **Example:** `Device(config-router)# exit`	Exits router configuration mode, and returns the device to global configuration mode.
Step 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map nh6 permit 10`	Defines a route map and enters route-map configuration mode.
Step 11	**match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*} **Example:** `Device(config-route-map)# match ipv6 address prefix-list cisco`	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.
Step 12	**set ipv6 next-hop** *ipv6-address* [*link-local-address*] [**peer-address**] **Example:** `Device(config-route-map)# set ipv6 next-hop 2001:DB8::1`	Overrides the next hop advertised to the peer for IPv6 packets that pass a match clause of a route map for policy routing. • The *ipv6-address* argument specifies the IPv6 global address of the next hop. It need not be an adjacent router. • The *link-local-address* argument specifies the IPv6 link-local address of the next hop. It must be an adjacent router. **Note** The route map sets the IPv6 next-hop addresses (global and link-local) in BGP updates. If the route map is not configured, the next-hop address in the BGP updates defaults to the unspecified IPv6 address (::), which is rejected by the peer.

Configuration Examples for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

Example: Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address

The following example configures the IPv6 multiprotocol BGP peer FE80::1234:BFF:FE0E:A471 over Gigabitethernet interface 0/0 and sets the route map named nh6 to include the IPv6 next-hop global address of Gigabitethernet interface 0/0 in BGP updates.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 5
Device(config-router)# neighbor internal peer-group
Device(config-router)# neighbor  FE80::1234:BFF:FE0E:A471% peer-group
Device(config-router)# neighbor internal remote-as 100
Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% remote-as 64600
Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471% update-source Gigabitethernet 0/0

Device(config-router)# address-family ipv6
Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% activate
Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471% route-map nh6 out
Device(config-router-af)# exit
Device(config-router)# exit
Device(config)# route-map nh6permit 10
Device(config-router-map)# match ipv6 address prefix-list cisco
Device(config-router-map)# set ipv6 next-hop 2001:DB8:526::1
Device(config-router-map)# exit
Device(config)# ipv6 prefix-list cisco permit 2001:DB8:2F22::/48 le 128
Device(config)# ipv6 prefix-list cisco deny ::/0
Device(config)# end
```

The following example configures the IPv6 multiprotocol BGP peer FE80::1234:BFF:FE0E:A471 over GigabitEthernetinterface 0/0/0 and sets the route map named nh6 to include the IPv6 next-hop global address of GigabitEthernet interface 0/0/0 in BGP updates.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 5
Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471%GigabitEthernet0/0/0 remote-as
64600
Device(config-router)# address-family ipv6
Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471%GigabitEthernet0/0/0 activate
Device(config-router-af)# neighbor FE80::1234:BFF:FE0E:A471%GigabitEthernet0/0/0 route-map
 nh6 out
Device(config-router-af)# exit

Device(config-router)# exit
Device(config)# route-map nh6permit 10
Device(config-router-map)# match ipv6 address prefix-list cisco
Device(config-router-map)# set ipv6 next-hop 2001:DB8:526::1
Device(config-router-map)# exit
Device(config)# ipv6 prefix-list cisco permit 2001:DB8:2F22::/48 le 128
```

```
Device(config)# ipv6 prefix-list cisco deny ::/0
Device(config)# end
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

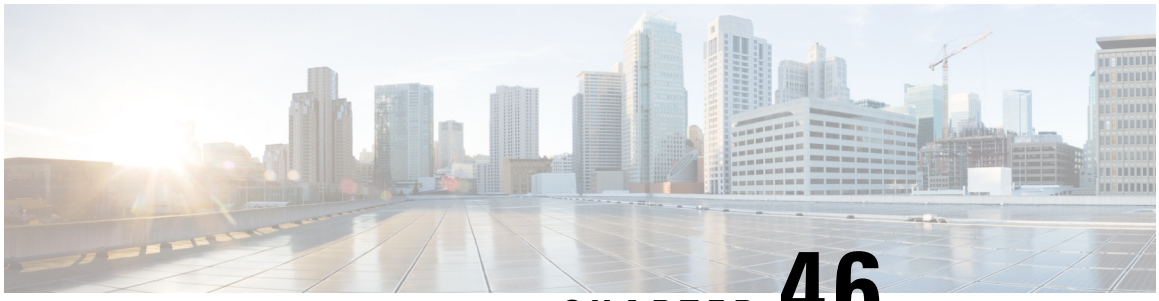
Feature Information for IPv6 Routing Multiprotocol BGP Link-Local Address Peering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 67: Feature Information for IPv6 Routing: Multiprotocol BGP Link-Local Address Peering

Feature Name	Releases	Feature Information
IPv6 Routing: Multiprotocol BGP Link-Local Address Peering	Cisco IOS XE Release 2.1	This feature is supported.



CHAPTER 46

IPv6 Multicast Address Family Support for Multiprotocol BGP

Information About IPv6 Multicast Address Family Support for Multiprotocol BGP

Multiprotocol BGP for the IPv6 Multicast Address Family

The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPv6 provides for interdomain transport of the same. Users must use multiprotocol BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality is provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are usable only for IP unicast, not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (for example, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

How to Implement IPv6 Multicast Address Family Support for Multiprotocol BGP

Configuring an IPv6 Peer Group to Perform Multicast BGP Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family ipv6** [**unicast** | **multicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified BGP routing process.
Step 4	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router)# neighbor group1 peer-group`	Creates a BGP peer group.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 2001:DB8:0:CC00::1 remote-as 64600`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multicast BGP neighbor table of the local router. • The *ipv6-address* argument in the **neighbor remote-as** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 6	**address-family ipv6** [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6 multicast`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified in the **address-family ipv6** command. • The **multicast** keyword specifies IPv6 multicast address prefixes.
Step 7	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate`	Enables the neighbor to exchange prefixes for the specified family type with the neighbor and the local router. • To avoid extra configuration steps for each neighbor, use the **neighbor activate** command with the *peer-group-name* argument as an alternative in this step.
Step 8	**neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 peer-group group1`	Assigns the IPv6 address of a BGP neighbor to a peer group.

Advertising Routes into IPv6 Multiprotocol BGP

By default, networks that are defined in router configuration mode using the **network** command are injected into the IPv4 unicast database. To inject a network into another database, such as the IPv6 BGP database, you must define the network using the **network** command in address family configuration mode for the other database, as shown for the IPv6 BGP database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
5. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified BGP routing process.
Step 4	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] **Example:** `Device(config-router)# address-family ipv6 unicast`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword is not specified with the **address-family ipv6** command. • The **multicast** keyword specifies IPv6 multicast address prefixes.
Step 5	**network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*] **Example:** `Device(config-router-af)# network 2001:DB8::/24`	Advertises (injects) the specified prefix into the IPv6 BGP database (the routes must first be found in the IPv6 unicast routing table). • The prefix is injected into the database for the address family specified in the previous step. • Routes are tagged from the specified prefix as "local origin." • The *ipv6-prefix* argument in the **network** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. • The *prefix-length* argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode, and returns the device to router configuration mode. • Repeat this step to exit router configuration mode and return the device to global configuration mode.

Redistributing Prefixes into IPv6 Multiprotocol BGP

Redistribution is the process of redistributing, or injecting, prefixes from one routing protocol into another routing protocol. This task explains how to inject prefixes from a routing protocol into IPv6 multiprotocol BGP. Specifically, prefixes that are redistributed into IPv6 multiprotocol BGP using the **redistribute** router configuration command are injected into the IPv6 unicast database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
5. **redistribute bgp** [*process-id*] [**metric** *metric-value*] [**route-map** *map-name*] [*source-protocol-options*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified BGP routing process.
Step 4	**address-family ipv6** [**vrf** *vrf-name*] [**unicast**	**multicast**	**vpnv6**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. By default, the device is placed in configuration mode for the IPv6 unicast address family if a keyword

	Command or Action	Purpose
		is not specified with the **address-family ipv6** command. • The **multicast** keyword specifies IPv6 multicast address prefixes.
Step 5	**redistribute bgp** [*process-id*] [**metric** *metric-value*] [**route-map** *map-name*] [*source-protocol-options*] **Example:** `Device(config-router-af)# redistribute bgp 64500 metric 5`	Redistributes IPv6 routes from one routing domain into another routing domain.
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode, and returns the device to router configuration mode. • Repeat this step to exit router configuration mode and return the device to global configuration mode.

Assigning a BGP Administrative Distance

Caution Changing the administrative distance of BGP internal routes is not recommended. One problem that can occur is the accumulation of routing table inconsistencies, which can break routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast** | **multicast**}
5. **distance bgp** *external-distance internal-distance local-distance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv6** [**unicast** | **multicast**} **Example:** `Device(config-router)# address-family ipv6 multicast`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 5	**distance bgp** *external-distance internal-distance local-distance* **Example:** `Device(config-router)# distance bgp 20 20 200`	Assigns a BGP administrative distance.

Generating Translate Updates for IPv6 Multicast BGP

The multicast BGP translate-update feature generally is used in a multicast BGP-capable router that peers with a customer site that has only a BGP-capable router; the customer site has not or cannot upgrade its router to a multicast BGP-capable image. Because the customer site cannot originate multicast BGP advertisements, the router with which it peers will translate the BGP prefixes into multicast BGP prefixes, which are used for multicast-source RPF lookup.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast** | **multicast**}
5. **neighbor** *ipv6-address* **translate-update ipv6 multicast** [**unicast**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv6** [**unicast** | **multicast**} **Example:** `Device(config-router)# address-family ipv6 multicast`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 5	**neighbor** *ipv6-address* **translate-update ipv6 multicast** [**unicast**] **Example:** `Device(config-router)# neighbor 2001:DB8:7000::2 translate-update ipv6 multicast`	Generates multiprotocol IPv6 BGP updates that correspond to unicast IPv6 updates received from a peer.

Resetting IPv6 BGP Sessions

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6** {**unicast** | **multicast**} {***** | *autonomous-system-number* | *ip-address* | *ipv6-address* | **peer-group** *peer-group-name*} [**soft**] [**in** | **out**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear bgp ipv6** {**unicast** | **multicast**} {***** | *autonomous-system-number* | *ip-address* | *ipv6-address* | **peer-group** *peer-group-name*} [**soft**] [**in** | **out**] **Example:** `Device# clear bgp ipv6 unicast peer-group marketing soft out`	Resets IPv6 BGP sessions.

Clearing External BGP Peers

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6** {**unicast** | **multicast**} **external** [**soft**] [**in** | **out**]
3. **clear bgp ipv6** {**unicast** | **multicast**} **peer-group** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear bgp ipv6** {**unicast**	**multicast**} **external** [**soft**] [**in**	**out**] **Example:** `Device# clear bgp ipv6 unicast external soft in`	Clears external IPv6 BGP peers.
Step 3	**clear bgp ipv6** {**unicast**	**multicast**} **peer-group** *name* **Example:** `Device# clear bgp ipv6 unicast peer-group marketing`	Clears all members of an IPv6 BGP peer group.

Clearing IPv6 BGP Route Dampening Information

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6** {**unicast** | **multicast**} **dampening** [*ipv6-prefix*/*prefix-length*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear bgp ipv6** {**unicast**	**multicast**} **dampening** [*ipv6-prefix*/*prefix-length*] **Example:**	Clears IPv6 BGP route dampening information and unsuppresses the suppressed routes.

	Command or Action	Purpose
	`Device# clear bgp ipv6 unicast dampening 2001:DB8::/64`	

Clearing IPv6 BGP Flap Statistics

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6** {**unicast** | **multicast**} **flap-statistics** [*ipv6-prefix/prefix-length* | **regexp** *regexp* | **filter-list** *list*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear bgp ipv6** {**unicast**	**multicast**} **flap-statistics** [*ipv6-prefix/prefix-length*	**regexp** *regexp*	**filter-list** *list*] **Example:** `Device# clear bgp ipv6 unicast flap-statistics filter-list 3`	Clears IPv6 BGP flap statistics.

Configuration Examples for IPv6 Multicast Address Family Support for Multiprotocol BGP

Example: Configuring an IPv6 Multiprotocol BGP Peer Group

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 65000
no bgp default ipv4-unicast
neighbor group1 peer-group
neighbor 2001:DB8:0:CC00::1 remote-as 64600
address-family ipv6 unicast
 neighbor group1 activate
 neighbor 2001:DB8:0:CC00::1 peer-group group1
```

Example: Advertising Routes into IPv6 Multiprotocol BGP

The following example injects the IPv6 network 2001:DB8::/24 into the IPv6 unicast database of the local device. (BGP checks that a route for the network exists in the IPv6 unicast database of the local device before advertising the network.)

```
router bgp 65000
 no bgp default ipv4-unicast
address-family ipv6 unicast
  network 2001:DB8::/24
```

Example: Redistributing Prefixes into IPv6 Multiprotocol BGP

The following example redistributes RIP routes into the IPv6 unicast database of the local device:

```
router bgp 64900
no bgp default ipv4-unicast
address-family ipv6 unicast
 redistribute rip
```

Example: Generating Translate Updates for IPv6 Multicast BGP

The following example shows how to generate IPv6 multicast BGP updates that correspond to unicast IPv6 updates:

```
router bgp 64900
 no bgp default ipv4-unicast
address-family ipv6 multicast
 neighbor 2001:DB8:7000::2 translate-update ipv6 multicast
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

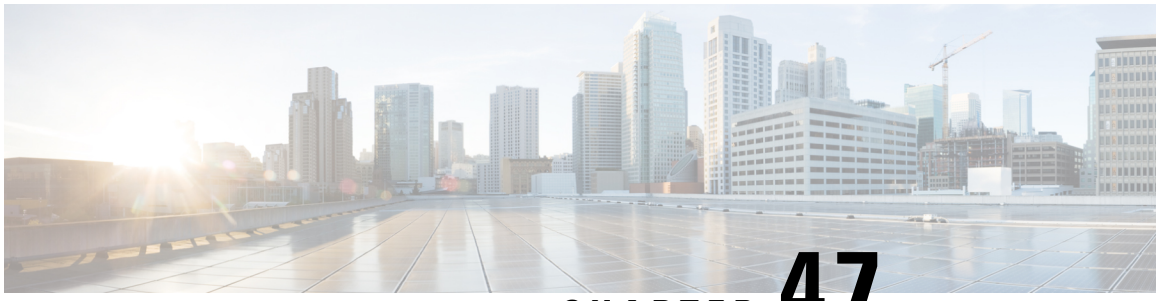
Feature Information for IPv6 Multicast Address Family Support for Multiprotocol BGP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 68: Feature Information for IPv6 Multicast: Address Family Support for Multiprotocol BGP

Feature Name	Releases	Feature Information
IPv6 Multicast: Address Family Support for Multiprotocol BGP	Cisco IOS XE Release 2.1	This feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP.



CHAPTER 47

Configuring Multiprotocol BGP (MP-BGP) Support for CLNS

This module describes configuration tasks to configure multiprotocol BGP (MP-BGP) support for CLNS, which provides the ability to scale Connectionless Network Service (CLNS) networks. The multiprotocol extensions of Border Gateway Protocol (BGP) add the ability to interconnect separate Open System Interconnection (OSI) routing domains without merging the routing domains, thus providing the capability to build very large OSI networks.

Restrictions for Configuring MP-BGP Support for CLNS

The configuration of MP-BGP support for CLNS does not support the creation and use of BGP confederations within the CLNS network. We recommend the use of route reflectors to address the issue of a large internal BGP mesh.

BGP extended communities are not supported by the MP-BGP Support for CLNS feature.

The following BGP commands are not supported by the MP-BGP Support for CLNS feature:

- **auto-summary**
- **neighbor advertise-map**
- **neighbor distribute-list**
- **neighbor soft-reconfiguration**
- **neighbor unsuppress-map**

Information About Configuring MP-BGP Support for CLNS

Address Family Routing Information

By default, commands entered under the **router bgp** command apply to the IPv4 address family. This will continue to be the case unless you enter the **no bgp default ipv4-unicast** command as the first command under the **router bgp** command. The **no bgp default ipv4-unicast** command is configured on the router to disable the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

Design Features of MP-BGP Support for CLNS

The configuration of MP-BGP support for CLNS allows BGP to be used as an interdomain routing protocol in networks that use CLNS as the network-layer protocol. This feature was developed to solve a scaling issue with a data communications network (DCN) where large numbers of network elements are managed remotely. For details about the DCN issues and how to implement this feature in a DCN topology, see the DCN Network Topology, on page 714.

BGP, as an Exterior Gateway Protocol, was designed to handle the volume of routing information generated by the Internet. Network administrators can control the BGP routing information because BGP neighbor relationships (peering) are manually configured and routing updates use incremental broadcasts. Some interior routing protocols such as Intermediate System-to-Intermediate System (IS-IS), in contrast, use a form of automatic neighbor discovery and broadcast updates at regular intervals.

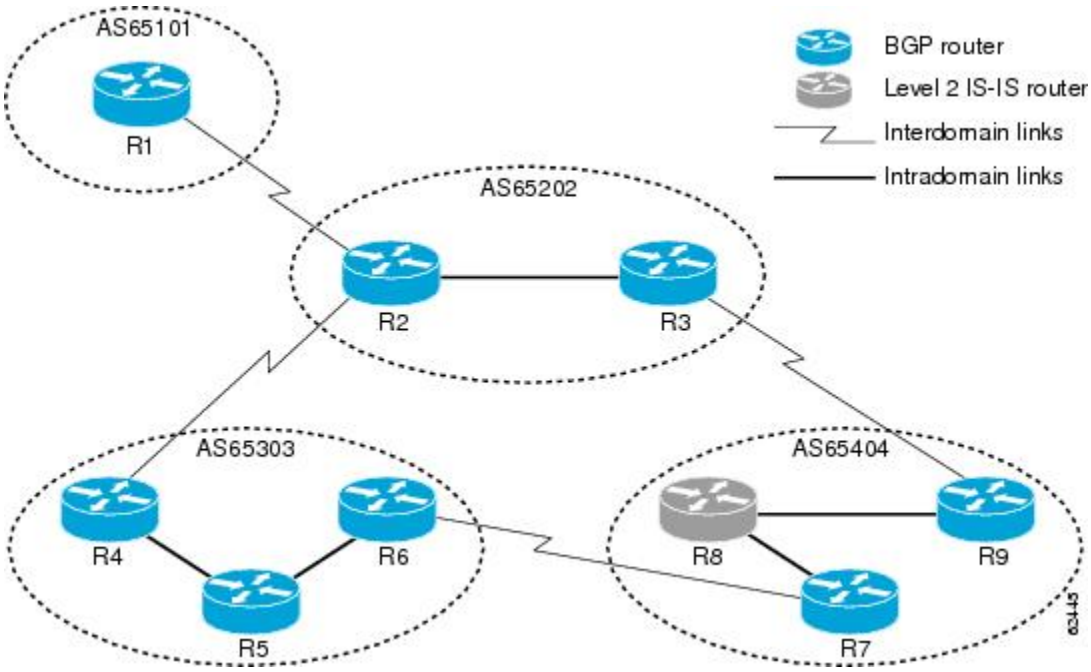
CLNS uses network service access point (NSAP) addresses to identify all its network elements. Using the BGP address-family support, NSAP address prefixes can be transported using BGP. In CLNS, BGP prefixes are inserted into the CLNS Level 2 prefix table. This functionality allows BGP to be used as an interdomain routing protocol between separate CLNS routing domains.

Implementing BGP in routers at the edge of each internal network means that the existing interior protocols need not be changed, minimizing disruption in the network.

Generic BGP CLNS Network Topology

The figure below shows a generic BGP CLNS network containing nine routers that are grouped into four different autonomous systems (in BGP terminology) or routing domains (in OSI terminology). To avoid confusion, we will use the BGP terminology of autonomous systems because each autonomous system is numbered and therefore more easily identified in the diagram and in the configuration discussion.

Figure 50: Components in a Generic BGP CLNS Network



Within each autonomous system, IS-IS is used as the intradomain routing protocol. Between autonomous systems, BGP and its multiprotocol extensions are used as the interdomain routing protocol. Each router is running either a BGP or Level 2 IS-IS routing process. To facilitate this feature, the BGP routers are also running a Level 2 IS-IS process. Although the links are not shown in the figure, each Level 2 IS-IS router is connected to multiple Level 1 IS-IS routers that are, in turn, connected to multiple CLNS networks.

Each autonomous system in this example is configured to demonstrate various BGP features and how these features work with CLNS to provide a scalable interdomain routing solution. In the figure above, the autonomous system AS65101 has a single Level 2 IS-IS router, R1, and is connected to just one other autonomous system, AS65202. Connectivity to the rest of the network is provided by R2, and a default route is generated for R1 to send to R2 all packets with destination NSAP addresses outside of AS65101.

In AS65202 there are two routers, R2 and R3, both with different external BGP (eBGP) neighbors. Routers R2 and R3 are configured to run internal BGP (iBGP) over the internal connection between them.

AS65303 shows how the use of BGP peer groups and route reflection can minimize the need for TCP connections between routers. Fewer connections between routers simplifies the network design and the amount of traffic in the network.

AS65404 shows how to use redistribution to communicate network reachability information to a Level 2 IS-IS router that is not running BGP.

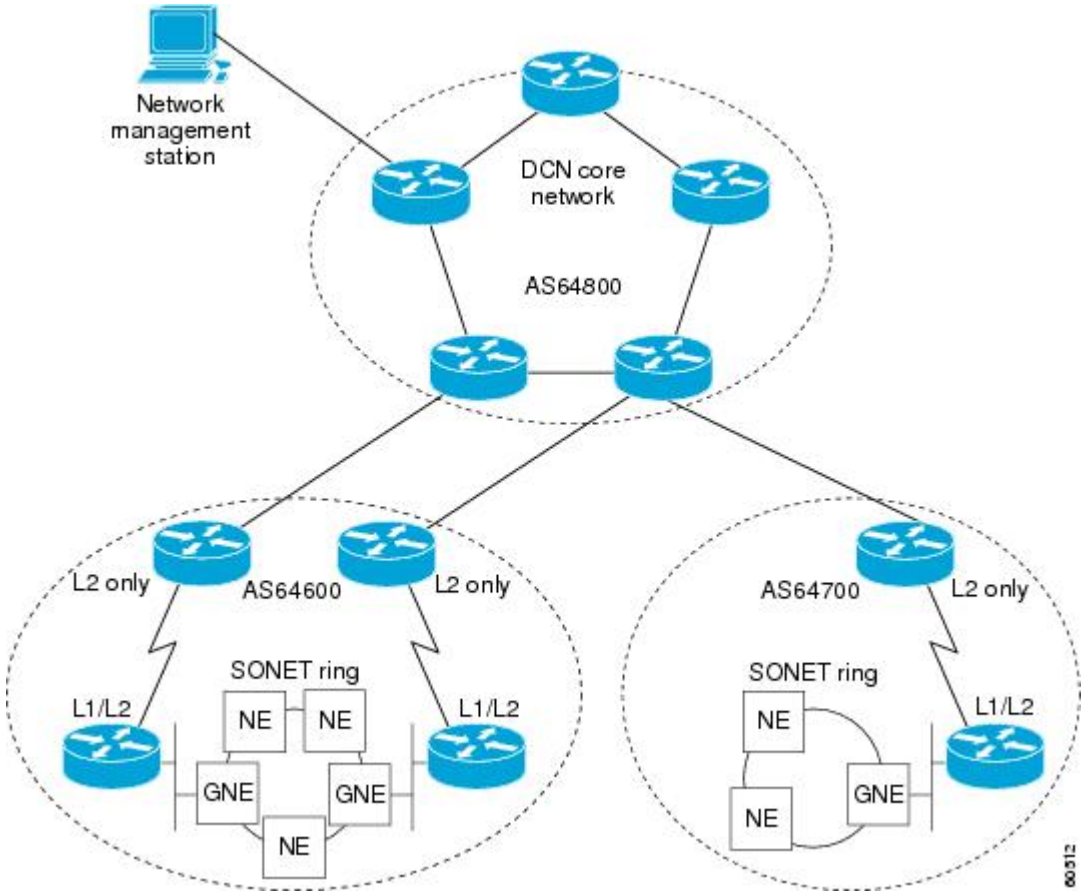
The configuration tasks and examples are based on the generic network design shown in the figure above. Configurations for all the routers in the figure are listed in the Implementing MP-BGP Support for CLNS Example, on page 739.

DCN Network Topology

The Multiprotocol BGP (MP-BGP) Support for CLNS feature can benefit a DCN managing a large number of remote SONET rings. SONET is typically used by telecommunications companies to send data over fiber-optic networks.

The figure below shows some components of a DCN network. To be consistent with the BGP terminology, the figure contains labels to indicate three autonomous systems instead of routing domains. The network elements--designated by NE in Figure 2--of a SONET ring are managed by OSI protocols such as File Transfer, Access, and Management (FTAM) and Common Management Information Protocol (CMIP). FTAM and CMIP run over the CLNS network-layer protocol, which means that the routers providing connectivity must run an OSI routing protocol.

Figure 51: Components in a DCN Network



IS-IS is a link-state protocol used in this example to route CLNS. Each routing node (networking device) is called an intermediate system (IS). The network is divided into areas defined as a collection of routing nodes. Routing within an area is referred to as Level 1 routing. Routing between areas involves Level 2 routing. Routers that link a Level 1 area with a Level 2 area are defined as Level 1-2 routers. A network element that connects to the Level 2 routers that provide a path to the DCN core is represented by a gateway network element--GNE in Figure 2. The network topology here is a point-to-point link between each network element router. In this example, a Level 1 IS-IS router is called an NE router.

Smaller Cisco routers such as the Cisco 2600 series were selected to run as the Level 1-2 routers because shelf space in the central office (CO) of a service provider is very expensive. A Cisco 2600 series router has limited processing power if it is acting as the Level 1 router for four or five different Level 1 areas. The number of Level 1 areas under this configuration is limited to about 200. The entire Level 2 network is also limited by the speed of the slowest Level 2 router.

To provide connectivity between NE routers, in-band signaling is used. The in-band signaling is carried in the SONET/Synchronous Digital Hierarchy (SDH) frame on the data communications channel (DCC). The DCC is a 192-KB channel, which is a very limited amount of bandwidth for the management traffic. Due to the limited signaling bandwidth between network elements and the limited amount of processing power and memory in the NE routers running IS-IS, each area is restricted to a maximum number of 30 to 40 routers. On average, each SONET ring consists of 10 to 15 network elements.

With a maximum of 200 areas containing 10 to 15 network elements per area, the total number of network element routers in a single autonomous system must be fewer than 3000. Service providers are looking to implement over 10,000 network elements as their networks grow, but the potential number of network elements in an area is limited. The current solution is to break down the DCN into a number of smaller autonomous systems and connect them using static routes or ISO Interior Gateway Routing Protocol (IGRP). ISO IGRP is a proprietary protocol that can limit future equipment implementation options. Static routing does not scale because the growth in the network can exceed the ability of a network administrator to maintain the static routes. BGP has been shown to scale to over 100,000 routes.

To implement the Multiprotocol BGP (MP-BGP) Support for CLNS feature in this example, configure BGP to run on each router in the DCN core network--AS64800 in Figure 2--to exchange routing information between all the autonomous systems. In the autonomous systems AS64600 and AS64700, only the Level 2 routers will run BGP. BGP uses TCP to communicate with BGP-speaking neighbor routers, which means that both an IP-addressed network and an NSAP-addressed network must be configured to cover all the Level 2 IS-IS routers in the autonomous systems AS64600 and AS64700 and all the routers in the DCN core network.

Assuming that each autonomous system--for example, AS64600 and AS64700 in Figure 2--remains the same size with up to 3000 nodes, we can demonstrate how large DCN networks can be supported with this feature. Each autonomous system advertises one address prefix to the core autonomous system. Each address prefix can have two paths associated with it to provide redundancy because there are two links between each autonomous system and the core autonomous system. BGP has been shown to support 100,000 routes, so the core autonomous system can support many other directly linked autonomous systems because each autonomous system generates only a few routes. We can assume that the core autonomous system can support about 2000 directly linked autonomous systems. With the hub-and-spoke design where each autonomous system is directly linked to the core autonomous system, and not acting as a transit autonomous system, the core autonomous system can generate a default route to each linked autonomous system. Using the default routes, the Level 2 routers in the linked autonomous systems process only a small amount of additional routing information. Multiplying the 2000 linked autonomous systems by the 3000 nodes within each autonomous system could allow up to 6 million network elements.

Benefits of MP-BGP Support for CLNS

The Multiprotocol BGP (MP-BGP) Support for CLNS feature adds the ability to interconnect separate OSI routing domains without merging the routing domains, which provides the capability to build very large OSI networks. The benefits of using this feature are not confined to DCN networks, and can be implemented to help scale any network using OSI routing protocols with CLNS.

How to Configure MP-BGP Support for CLNS

Configuring and Activating a BGP Neighbor to Support CLNS

To configure and activate a BGP routing process and an associated BGP neighbor (peer) to support CLNS, perform the steps in this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family nsap** [**unicast**]
7. **neighbor** *ip-address* **activate**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65101`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument identifies the autonomous system in which the router resides. Valid values are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.1.2.2 remote-as 64202`	Adds an IP address or peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.

Step 6	**address-family nsap** [**unicast**]
	Example:
	`Router(config-router)#`
	`address-family nsap`
		Specifies the NSAP address family and enters address family configuration mode.
		• The optional **unicast** keyword specifies the NSAP unicast address prefixes. By default, the router is placed in configuration mode for the unicast NSAP address family if the **unicast** keyword is not specified with the **address-family nsap** command.

Step 7	**neighbor** *ip-address* **activate**
	Example:
	`Router(config-router-af)#`
	`neighbor 10.1.2.2 activate`
		Enables the BGP neighbor to exchange prefixes for the NSAP address family with the local router.
		Note If you have configured a peer group as a BGP neighbor, you do not use this command because peer groups are automatically activated when any peer group parameter is configured.

Step 8	**end**
	Example:
	`Router(config-router-af)#`
	`end`
		Exits address family configuration mode and returns to privileged EXEC mode.

Configuring an IS-IS Routing Process

When an integrated IS-IS routing process is configured, the first instance of the IS-IS routing process configured is by default a Level 1-2 (intra-area and interarea) router. All subsequent IS-IS routing processes on a network running CLNS are configured as Level 1. All subsequent IS-IS routing processes on a network running IP are configured as Level-1-2. To use the Multiprotocol BGP (MP-BGP) Support for CLNS feature, configure a Level 2 routing process.

To configure an IS-IS routing process and assign it as a Level-2-only process, perform the steps in this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **net** *network-entity-title*
5. **is-type** [**level-1** | **level-1-2** | **level-2-only**]

6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis osi-as-101`	Configures an IS-IS routing process and enters router configuration mode for the specified routing process. • The *area-tag* argument is a meaningful name for a routing process. It must be unique among all IP and CLNS routing processes for a given router.
Step 4	**net** *network-entity-title* **Example:** `Router(config-router)# net 49.0101.1111.1111.1111.1111.00`	Configures a network entity title (NET) for the routing process. • If you are configuring multiarea IS-IS, you must specify a NET for each routing process.
Step 5	**is-type** [**level-1** | **level-1-2** | **level-2-only**] **Example:** `Router(config-router)# is-type level-1`	Configures the router to act as a Level 1 (intra-area) router, as both a Level 1 router and a Level 2 (interarea) router, or as an interarea router only. • In multiarea IS-IS configurations, the first instance of the IS-IS routing process configured is by default a Level 1-2 (intra-area and interarea) router. All subsequent IS-IS routing processes on a network running CLNS are configured as Level 1. All subsequent IS-IS routing processes on a network running IP are configured as Level 1-2.

Step 6	**end**
	Example:
	`Router(config-router)#`
	`end`
		Exits router configuration mode and returns to privileged EXEC mode.

Configuring Interfaces That Connect to BGP Neighbors

When a router running IS-IS is directly connected to an eBGP neighbor, the interface between the two eBGP neighbors is activated using the **clns enable** command, which allows CLNS packets to be forwarded across the interface. The **clns enable** command activates the End System-to-Intermediate System (ES-IS) protocol to search for neighboring OSI systems.

Note Running IS-IS across the same interface that is connected to an eBGP neighbor can lead to undesirable results if the two OSI routing domains merge into a single domain.

When a neighboring OSI system is found, BGP checks that it is also an eBGP neighbor configured for the NSAP address family. If both the preceding conditions are met, BGP creates a special BGP neighbor route in the CLNS Level 2 prefix routing table. The special BGP neighbor route is automatically redistributed in to the Level 2 routing updates so that all other Level 2 IS-IS routers in the local OSI routing domain know how to reach this eBGP neighbor.

To configure interfaces that are being used to connect with eBGP neighbors, perform the steps in this procedure. These interfaces will normally be directly connected to their eBGP neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **clns enable**
6. **no shutdown**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface serial 2/0/0`	Specifies the interface type and number and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.2.2 255.255.255.0`	Configures the interface with an IP address.

Step 5	**clns enable**
	Example:
	`Router(config-if)#`
	`clns enable`
		Specifies that CLNS packets can be forwarded across this interface.
		• The ES-IS protocol is activated and starts to search for adjacent OSI systems.

Step 6	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Turns on the interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Interfaces Connected to the Local OSI Routing Domain

To configure interfaces that are connected to the local OSI routing domain, perform the steps in this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **clns router isis** *area-tag*
6. **ip router isis** *area-tag*
7. **no shutdown**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.2.3.1 255.255.255.0`	Configures the interface with an IP address. **Note** This step is required only when the interface needs to communicate with an iBGP neighbor.

Step 5	**clns router isis** *area-tag*
	Example:
	`Router(config-if)#`
	`clns router isis osi-as-202`
		Specifies that the interface is actively routing IS-IS when the network protocol is ISO CLNS and identifies the area associated with this routing process.

Step 6	**ip router isis** *area-tag*
	Example:
	`Router(config-if)#`
	`ip router isis`
	`osi-as-202`
		Specifies that the interface is actively routing IS-IS when the network protocol is IP and identifies the area associated with this routing process.
		Note This step is required only when the interface needs to communicate with an iBGP neighbor, and the IGP is IS-IS.

Step 7	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Turns on the interface.
Step 8	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Advertising Networking Prefixes

Advertising NSAP address prefix forces the prefixes to be added to the BGP routing table. To configure advertisement of networking prefixes, perform the steps in this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family nsap** [**unicast**]
7. **network** *nsap-prefix* [**route-map** *map-tag*]
8. **neighbor** *ip-address* **activate**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65101`	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.1.2.2 remote-as 64202`	Adds an IP address or peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.
Step 6	**address-family nsap** [**unicast**] **Example:** `Router(config-router)# address-family nsap`	Specifies the NSAP address family and enters address family configuration mode. • The optional **unicast** keyword specifies the NSAP unicast address prefixes. By default, the router is placed in unicast NSAP address family configuration mode if the **unicast** keyword is not specified with the **address-family nsap** command.

	Command or Action	Purpose
Step 7	**network** *nsap-prefix* [**route-map** *map-tag*] **Example:** `Router(config-router-af)# network 49.0101.1111.1111.1111.1111.00`	Advertises a single prefix of the local OSI routing domain and enters it in the BGP routing table. **Note** It is possible to advertise a single prefix, in which case this prefix could be the unique NSAP address prefix of the local OSI routing domain. Alternatively, multiple longer prefixes, each covering a small portion of the OSI routing domain, can be used to selectively advertise different areas. • The advertising of NSAP address prefixes can be controlled by using the optional **route-map** keyword. If no route map is specified, all NSAP address prefixes are redistributed.
Step 8	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af) neighbor 10.1.2.2 activate`	Specifies that NSAP routing information will be sent to the specified BGP neighbor. **Note** See the description of the **neighbor** command in the documents listed in the "Additional References" for more details on the use of this command.
Step 9	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Redistributing Routes from BGP into IS-IS

Route redistribution must be approached with caution. We do not recommend injecting the full set of BGP routes into IS-IS because excessive routing traffic will be added to IS-IS. Route maps can be used to control which dynamic routes are redistributed.

To configure route redistribution from BGP into IS-IS, perform the steps in this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **net** *network-entity-title*
5. **redistribute** *protocol as-number* [*route-type*] [**route-map** *map-tag*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis osi-as-404`	Configures an IS-IS routing process and enters router configuration mode for the specified routing process. **Note** You cannot redistribute BGP routes into a Level 1-only IS-IS routing process.
Step 4	**net** *network-entity-title* **Example:** `Router(config-router)# net 49.0404.7777.7777.7777.7777.00`	Configures a NET for the routing process. • If you are configuring multiarea IS-IS, you must specify a NET for each routing process.
Step 5	**redistribute** *protocol as-number* [*route-type*] [**route-map** *map-tag*] **Example:** `Router(config-router)# redistribute bgp 65404 clns`	Redistributes NSAP prefix routes from BGP into the CLNS Level 2 routing table associated with the IS-IS routing process when the *protocol* argument is set to **bgp**and the *route-type* argument is set to **clns**. • The *as-number* argument is defined as the autonomous system number of the BGP routing process to be redistributed into CLNS. • The redistribution of routes can be controlled by using the optional **route-map** keyword. If no route map is specified, all BGP routes are redistributed.

Step 6	**end**
	Example:
	`Router(config-router)#`
	`end`
		Exits router configuration mode and returns to privileged EXEC mode.

Redistributing Routes from IS-IS into BGP

Route redistribution must be approached with caution because redistributed route information is stored in the routing tables. Large routing tables may make the routing process slower. Route maps can be used to control which dynamic routes are redistributed.

To configure route redistribution from IS-IS into BGP, perform the steps in this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **address-family nsap** [**unicast**]
6. **redistribute** *protocol* [*process-id*] [*route-type*] [**route-map** *map-tag*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65202`	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

Step 5	**address-family nsap** [**unicast**]
	Example:
	`Router(config-router)#`
	`address-family nsap`
		Specifies the NSAP address family and enters address family configuration mode.

Step 6	**redistribute** *protocol* [*process-id*] [*route-type*] [**route-map** *map-tag*] **Example:** `Router(config-router-af)# redistribute isis osi-as-202 clns route-map internal-routes-only`	Redistributes routes from the CLNS Level 2 routing table associated with the IS-IS routing process into BGP as NSAP prefixes when the *protocol* argument is set to **isis**and the *route-type* argument is set to **clns**. • The *process-id* argument is defined as the area name for the relevant IS-IS routing process to be redistributed.

	Command or Action	Purpose
		• The redistribution of routes can be controlled by using the optional **route-map** keyword. If no route map is specified, all Level 2 routes are redistributed.

Step 7	**end**
	Example:
	`Router(config-router-af)#`
	`end`
		Exits address family configuration mode and returns to privileged EXEC mode.

Configuring BGP Peer Groups and Route Reflectors

BGP peer groups reduce the number of configuration commands by applying a BGP **neighbor** command to multiple neighbors. Using a BGP peer group with a local router configured as a BGP route reflector allows BGP routing information received from one member of the group to be replicated to all other group members. Without a peer group, each route reflector client must be specified by IP address.

To create a BGP peer group and use the group as a BGP route reflector client, perform the steps in this procedure. This is an optional task and is used with internal BGP neighbors. In this task, some of the BGP syntax is shown with the *peer-group-name* argument only and only one neighbor is configured as a member of the peer group. Repeat Step 9 to configure other BGP neighbors as members of the peer group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** *peer-group-name* **remote-as** *as-number*
7. **address-family nsap** [**unicast**]
8. **neighbor** *peer-group-name* **route-reflector-client**
9. **neighbor** *ip-address* **peer-group** *peer-group*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65303`	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `Router(config-router)# neighbor ibgp-peers peer-group`	Creates a BGP peer group.
Step 6	**neighbor** *peer-group-name* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor ibgp-peers remote-as 65303`	Adds the peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.

Step 7	**address-family nsap** [**unicast**]
	Example:
	`Router(config-router)#`
	`address-family nsap`
		Specifies the NSAP address family and enters address family configuration mode.

Step 8	**neighbor** *peer-group-name* **route-reflector-client**
	Example:
	`Router(config-router-af)#`
	`neighbor ibgp-peers route-reflector-client`
		Configures the router as a BGP route reflector and configures the specified peer group as its client.

Step 9	**neighbor** *ip-address* **peer-group** *peer-group*
	Example:
	`Router(config-router-af)#`
	`neighbor 10.4.5.4 peer-group ibgp-peers`
		Assigns a BGP neighbor to a BGP peer group.

Step 10	**end** **Example:** `Router(config-router-af)#`	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`end`	

Filtering Inbound Routes Based on NSAP Prefixes

Perform this task to filter inbound BGP routes based on NSAP prefixes. The **neighbor prefix-list in** command is configured in address family configuration mode to filter inbound routes.

Before you begin

You must specify either a CLNS filter set or a CLNS filter expression before configuring the **neighbor** command. See descriptions for the **clns filter-expr** and **clns filter-set** commands for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **address-family nsap** [**unicast**]
6. **neighbor** {*ip-address*| *peer-group-name*}**prefix-list** {*clns-filter-expr-name*| *clns-filter-set-name*} **in**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65200`	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

	Command or Action	Purpose

Step 5	**address-family nsap** [**unicast**]
	Example:
	`Router(config-router)#`
	`address-family nsap`
		Specifies the address family and enters address family configuration mode.

Step 6	**neighbor** {*ip-address*| *peer-group-name*}**prefix-list** {*clns-filter-expr-name*| *clns-filter-set-name*} **in**
	Example:
	`Router(config-router-af)#`
	`neighbor 10.23.4.1 prefix-list abc in`
		Specifies a CLNS filter set or CLNS filter expression to be used to filter inbound BGP routes.
		• The *clns-filter-expr-name* argument is defined with the **clns filter-expr** configuration command.
		• The *clns-filter-set-name* argument is defined with the **clns filter-set** configuration command.

Step 7	**end**
	Example:
	`Router(config-router-af)#`
	`end`
		Exits address family configuration mode and returns to privileged EXEC mode.

Filtering Outbound BGP Updates Based on NSAP Prefixes

Perform this task to filter outbound BGP updates based on NSAP prefixes, use the **neighbor prefix-list out** command in address family configuration mode. This task is configured at Router 7 in the figure above (in the "Generic BGP CLNS Network Topology" section). In this task, a CLNS filter is created with two entries to deny NSAP prefixes starting with 49.0404 and to permit all other NSAP prefixes starting with 49. A BGP peer group is created and the filter is applied to outbound BGP updates for the neighbor that is a member of the peer group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clns filter-set** *name* [**deny**] **template**
4. **clns filter-set** *name* [**permit**] **template**
5. **router bgp** *as-number*
6. **no bgp default ipv4-unicast**
7. **neighbor** *peer-group-name* **peer-group**
8. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
9. **address-family nsap** [**unicast**]
10. **neighbor** {*ip-address* | *peer-group-name*} **prefix-list** {*clns-filter-expr-name* | *clns-filter-set-name*} **out**
11. **neighbor** *ip-address* **peer-group** *peer-group*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**clns filter-set** *name* [**deny**] **template** **Example:** `Router(config)# clns filter-set routes0404 deny 49.0404...`	Defines a NSAP prefix match for a deny condition for use in CLNS filter expressions. • In this example, a deny action is returned if an address starts with 49.0404.
Step 4	**clns filter-set** *name* [**permit**] **template** **Example:** `Router(config)# clns filter-set routes0404 permit 49...`	Defines a NSAP prefix match for a permit condition for use in CLNS filter expressions. • In this example, a permit action is returned if an address starts with 49. **Note** Although the permit example in this step allows all NSAP addresses starting with 49, the match condition in Step 3 is processed first so the NSAP addresses starting with 49.0404 are still denied.
Step 5	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65404`	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 6	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.
Step 7	**neighbor** *peer-group-name* **peer-group** **Example:** `Router(config-router)# neighbor ebgp-peers peer-group`	Creates a BGP peer group. • In this example, the BGP peer group named ebgp-peers is created.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:**	Adds an IP address or peer group name of the BGP neighbor in the specified autonomous system to the BGP neighbor table of the local router.

	Command or Action	Purpose
	`Router(config-router)# neighbor ebgp-peers remote-as 65303`	• In this example, the peer group named ebgp-peers is added to the BGP neighbor table.

Step 9	**address-family nsap** [**unicast**]
	Example:
	`Router(config-router)#`
	`address-family nsap`
		Specifies the NSAP address family and enters address family configuration mode.

Step 10	**neighbor** {*ip-address* | *peer-group-name*} **prefix-list** {*clns-filter-expr-name* | *clns-filter-set-name*} **out**
	Example:
	`Router(config-router-af)#`
	`neighbor ebgp-peers prefix-list routes0404 out`
		Specifies a CLNS filter set or CLNS filter expression to be used to filter outbound BGP updates.
		• The *clns-filter-expr-name* argument is defined with the **clns filter-expr** configuration command.
		• The *clns-filter-set-name* argument is defined with the **clns filter-set** configuration command.
		• In this example, the filter set named routes0404 was created in Step3 and Step 4.

Step 11	**neighbor** *ip-address* **peer-group** *peer-group*
	Example:
	`Router(config-router-af)#`
	`neighbor 10.6.7.8 peer-group ebgp-peers`
		Assigns a BGP neighbor to a BGP peer group.

Step 12	**end**
	Example:
	`Router(config-router-af)#`
	`end`
		Exits address family configuration mode and returns to privileged EXEC mode.

Originating Default Routes for a Neighboring Routing Domain

To create a default CLNS route that points to the local router on behalf of a neighboring OSI routing domain, perform the steps in this procedure. This is an optional task and is normally used only with external BGP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **address-family nsap** [**unicast**]

6. **neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** map-tag]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 64803`	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

Step 5	**address-family nsap** [**unicast**]
	Example:
	`Router(config-router)#`
	`address-family nsap`
		Specifies the NSAP address family and enters address family configuration mode.

Step 6	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** map-tag]
	Example:
	`Router(config-router-af)#`
	` neighbor 172.16.2.3 default-originate`
		Generates a default CLNS route that points to the local router and that will be advertised to the neighboring OSI routing domain.

Step 7	**end**
	Example:
	`Router(config-router-af)#`
	`end`
		Exits address family configuration mode and returns to privileged EXEC mode.

Verifying MP-BGP Support for CLNS

To verify the configuration, use the **show running-config** EXEC command. Sample output is located in the Implementing MP-BGP Support for CLNS Example, on page 739. To verify that the Multiprotocol BGP (MP-BGP) Support for CLNS feature is working, perform the following steps.

SUMMARY STEPS

1. **show clns neighbors**
2. **show clns route**
3. **show bgp nsap unicast summary**
4. **show bgp nsap unicast**

DETAILED STEPS

Step 1 show clns neighbors

Use this command to confirm that the local router has formed all the necessary IS-IS adjacencies with other Level 2 IS-IS routers in the local OSI routing domain. If the local router has any directly connected external BGP peers, the output from this command will show that the external neighbors have been discovered, in the form of ES-IS adjacencies.

In the following example, the output is displayed for router R2, shown in the figure above (in the "Generic BGP CLNS Network Topology" section). R2 has three CLNS neighbors. R1 and R4 are ES-IS neighbors because these nodes are in different autonomous systems from R2. R3 is an IS-IS neighbor because it is in the same autonomous system as R2. Note that the system ID is replaced by CLNS hostnames (r1, r3, and r4) that are defined at the start of each configuration file. Specifying the CLNS hostname means that you need not remember which system ID corresponds to which hostname.

Example:

```
Router# show clns neighbors
Tag osi-as-202:
System Id      Interface   SNPA            State  Holdtime  Type Protocol
r1             Se2/0       *HDLC*          Up     274       IS   ES-IS
r3             Et0/1       0002.16de.8481  Up     9         L2   IS-IS
r4             Se2/2       *HDLC*          Up     275       IS   ES-IS
```

Step 2 show clns route

Use this command to confirm that the local router has calculated routes to other areas in the local OSI routing domain. In the following example of output from router R2, shown in the figure above (in the "Generic BGP CLNS Network Topology" section), the routing table entry--i 49.0202.3333 [110/10] via R3--shows that router R2 knows about other local IS-IS areas within the local OSI routing domain.

Example:

```
Router# show clns route
Codes: C - connected, S - static, d - DecnetIV
       I - ISO-IGRP,  i - IS-IS,  e - ES-IS
       B - BGP,       b - eBGP-neighbor
C  49.0202.2222 [2/0], Local IS-IS Area
C  49.0202.2222.2222.2222.2222.00 [1/0], Local IS-IS NET
b  49.0101.1111.1111.1111.1111.00 [15/10]
      via r1, Serial2/0
i  49.0202.3333 [110/10]
      via r3, GigabitEthernet0/1/1
```

```
b  49.0303.4444.4444.4444.4444.00 [15/10]
      via r4, Serial2/2
B  49.0101 [20/1]
      via r1, Serial2/0
B  49.0303 [20/1]
      via r4, Serial2/2
B  49.0404 [200/1]
      via r9
i  49.0404.9999.9999.9999.9999.00 [110/10]
      via r3, GigabitEthernet0/1/1
```

Step 3 show bgp nsap unicast summary

Use this command to verify that the TCP connection to a particular neighbor is active. In the following example output, search the appropriate row based on the IP address of the neighbor. If the State/PfxRcd column entry is a number, including zero, the TCP connection for that neighbor is active.

Example:

```
Router# show bgp nsap unicast summary
BGP router identifier 10.1.57.11, local AS number 65202
BGP table version is 6, main routing table version 6
5 network entries and 8 paths using 1141 bytes of memory
6 BGP path attribute entries using 360 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 5/0 prefixes, 8/0 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.2.1        4 65101      34      34        6    0    0 00:29:11        1
10.2.3.3        4 65202      35      36        6    0    0 00:29:16        3
```

Step 4 show bgp nsap unicast

Enter the **show bgp nsap unicast** command to display all the NSAP prefix routes that the local router has discovered. In the following example of output from router R2, shown in the figure above (in the "Generic BGP CLNS Network Topology" section), a single valid route to prefix 49.0101 is shown. Two valid routes--marked by a *--are shown for the prefix 49.0404. The second route is marked with a *>i sequence, representing the best route to this prefix.

Example:

```
Router# show bgp nsap unicast
BGP table version is 3, local router ID is 192.168.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 49.0101          49.0101.1111.1111.1111.1111.00
                                                          0 65101 i
* i49.0202.2222     49.0202.3333.3333.3333.3333.00
                                               100      0 ?
*>                  49.0202.2222.2222.2222.2222.00
                                                      32768 ?
* i49.0202.3333     49.0202.3333.3333.3333.3333.00
                                               100      0 ?
*>                  49.0202.2222.2222.2222.2222.00
                                                      32768 ?
*> 49.0303          49.0303.4444.4444.4444.4444.00
                                                          0 65303 i
*  49.0404          49.0303.4444.4444.4444.4444.00
                                                          0 65303 65404 i
```

```
*>i                49.0404.9999.9999.9999.9999.00
                                                100      0 65404 i
```

Troubleshooting MP-BGP Support for CLNS

The **debug bgp nsap unicast**commands enable diagnostic output concerning various events relating to the operation of the CLNS packets in the BGP routing protocol to be displayed on a console. These commands are intended only for troubleshooting purposes because the volume of output generated by the software when they are used can result in severe performance degradation on the router. See the *Cisco IOS Debug Command Reference* for more information about using these **debug** commands.

To troubleshoot problems with the configuration of MP-BGP support for CLNS and to minimize the impact of the **debug**commands used in this procedure, perform the following steps.

SUMMARY STEPS

1. Attach a console directly to a router running the Cisco software release that includes the Multiprotocol BGP (MP-BGP) Support for CLNS feature.
2. **no logging console**
3. Use Telnet to access a router port.
4. **enable**
5. **terminal monitor**
6. **debug bgp nsap unicast** [*neighbor-address* | **dampening** | **keepalives** | **updates**]
7. **no terminal monitor**
8. **no debug bgp nsap unicast** [*neighbor-address* | **dampening** | **keepalives** | **updates**]
9. **logging console**

DETAILED STEPS

Step 1 Attach a console directly to a router running the Cisco software release that includes the Multiprotocol BGP (MP-BGP) Support for CLNS feature.

Note This procedure will minimize the load on the router created by the **debug bgp nsap unicast** commands because the console port will no longer be generating character-by-character processor interrupts. If you cannot connect to a console directly, you can run this procedure via a terminal server. If you must break the Telnet connection, however, you may not be able to reconnect because the router may be unable to respond due to the processor load of generating the **debug bgp nsap unicast** output.

Step 2 **no logging console**

This command disables all logging to the console terminal.

Step 3 Use Telnet to access a router port.

Step 4 **enable**

Enter this command to access privileged EXEC mode.

Step 5 **terminal monitor**

This command enables logging on the virtual terminal.

Step 6 **debug bgp nsap unicast** [*neighbor-address* | **dampening** | **keepalives** | **updates**]

Enter only specific **debug bgp nsap unicast**commands to isolate the output to a certain subcomponent and minimize the load on the processor. Use appropriate arguments and keywords to generate more detailed debug information on specified subcomponents.

Step 7 **no terminal monitor**

This command disables logging on the virtual terminal.

Step 8 **no debug bgp nsap unicast** [*neighbor-address* | **dampening** | **keepalives** | **updates**]

Enter the specific **no debug bgp nsap unicast**command when you are finished.

Step 9 **logging console**

This command reenables logging to the console.

Configuration Examples for MP-BGP Support for CLNS

Example: Configuring and Activating a BGP Neighbor to Support CLNS

In the following example, the router R1, shown in the figure below, in the autonomous system AS65101 is configured to run BGP and activated to support CLNS. Router R1 is the only Level 2 IS-IS router in autonomous system AS65101, and it has only one connection to another autonomous system via router R2 in AS65202. The **no bgp default ipv4-unicast** command is configured on the router to disable the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers. After the NSAP address family configuration mode is enabled with the **address-family nsap** command, the router is configured to advertise the NSAP prefix of 49.0101 to its BGP neighbors and to send NSAP routing information to the BGP neighbor at 10.1.2.2.

```
router bgp 65101
 no bgp default ipv4-unicast
 address-family nsap
  network 49.0101...
  neighbor 10.1.2.2 activate
  exit-address-family
```

Example: Configuring an IS-IS Routing Process

In the following example, R1, shown in the figure below, is configured to run an IS-IS process:

```
router isis osi-as-101
 net 49.0101.1111.1111.1111.1111.00
```

The default IS-IS routing process level is used.

Configuring Interfaces Example

In the following example, two of the interfaces of the router R2, shown in the figure below, in the autonomous system AS65202 are configured to run CLNS. GigabitEthernet interface 0/1/1 is connected to the local OSI routing domain and is configured to run IS-IS when the network protocol is CLNS using the **clns router isis** command. The serial interface 2/0 with the local IP address of 10.1.2.2 is connected with an eBGP neighbor and is configured to run CLNS through the **clns enable** command:

```
interface serial 2/0
 ip address 10.1.2.2 255.255.255.0
 clns enable
 no shutdown
!
interface gigabitethernet 0/1/1
 ip address 10.2.3.1 255.255.255.0
 clns router isis osi-as-202
 no shutdown
```

Advertising Networking Prefixes Example

In the following example, the router R1, shown in the figure below, is configured to advertise the NSAP prefix of 49.0101 to other routers. The NSAP prefix unique to autonomous system AS65101 is advertised to allow the other autonomous systems to discover the existence of autonomous system AS65101 in the network.

```
router bgp 65101
 no bgp default ipv4-unicast
 neighbor 10.1.2.2 remote-as 64202
 address-family nsap
  network 49.0101...
  neighbor 10.1.2.2 activate
```

Example: Redistributing Routes from BGP into IS-IS

In the following example, the routers R7 and R9, shown in the figure below, in the autonomous system AS65404 are configured to redistribute BGP routes into the IS-IS routing process called osi-as-404. Redistributing the BGP routes allows the Level 2 IS-IS router, R8, to advertise routes to destinations outside the autonomous system AS65404. Without a route map being specified, all BGP routes are redistributed.

Router R7

```
router isis osi-as-404
 net 49.0404.7777.7777.7777.7777.00
 redistribute bgp 65404 clns
```

Router R9

```
router isis osi-as-404
 net 49.0404.9999.9999.9999.9999.00
 redistribute bgp 65404 clns
```

Example: Redistributing Routes from IS-IS into BGP

In the following example, the router R2, shown in the figure below, in the autonomous system AS65202 is configured to redistribute Level 2 CLNS NSAP routes into BGP. A route map is used to permit only routes from within the local autonomous system to be redistributed into BGP. Without a route map being specified, every NSAP route from the CLNS level 2 prefix table is redistributed. The **no bgp default ipv4-unicast** command is configured on the router to disable the default behavior of the BGP routing process exchanging IPv4 addressing information with BGP neighbor routers.

```
clns filter-set internal-routes permit 49.0202...
!
route-map internal-routes-only permit 10
 match clns address internal-routes
!
router isis osi-as-202
 net 49.0202.2222.2222.2222.2222.00
!
router bgp 65202
 no bgp default ipv4-unicast
 address-family nsap
 redistribute isis osi-as-202 clns route-map internal-routes-only
```

Configuring BGP Peer Groups and Route Reflectors Example

Router R5, shown in the figure above (in the "Generic BGP CLNS Network Topology" section), has only iBGP neighbors and runs IS-IS on both interfaces. To reduce the number of configuration commands, configure R5 as a member of a BGP peer group called ibgp-peers. The peer group is automatically activated under the **address-family nsap** command by configuring the peer group as a route reflector client allowing it to exchange NSAP routing information between group members. The BGP peer group is also configured as a BGP route reflector client to reduce the need for every iBGP router to be linked to each other.

In the following example, the router R5 in the autonomous system AS65303 is configured as a member of a BGP peer group and a BGP route reflector client:

```
router bgp 65303
 no bgp default ipv4-unicast
 neighbor ibgp-peers peer-group
 neighbor ibgp-peers remote-as 65303
 address-family nsap
  neighbor ibgp-peers route-reflector-client
  neighbor 10.4.5.4 peer-group ibgp-peers
  neighbor 10.5.6.6 peer-group ibgp-peers
  exit-address-family
```

Filtering Inbound Routes Based on NSAP Prefixes Example

In the following example, the router R1, shown in the figure below, in the autonomous system AS65101 is configured to filter inbound routes specified by the default-prefix-only prefix list:

```
clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
router isis osi-as-101
 net 49.0101.1111.1111.1111.1111.00
!
```

```
router bgp 65101
 no bgp default ipv4-unicast
 neighbor 10.1.2.2 remote-as 64202
 address-family nsap
  network 49.0101.1111.1111.1111.1111.00
  neighbor 10.1.2.2 activate
  neighbor 10.1.2.2 prefix-list default-prefix-only in
```

Example: Filtering Outbound BGP Updates Based on NSAP Prefixes

In the following example, outbound BGP updates are filtered based on NSAP prefixes. This example is configured at Router 7 in the figure below. In this task, a CLNS filter is created with two entries to deny NSAP prefixes starting with 49.0404 and to permit all other NSAP prefixes starting with 49. A BGP peer group is created and the filter is applied to outbound BGP updates for the neighbor that is a member of the peer group.

```
clns filter-set routes0404 deny 49.0404...
clns filter-set routes0404 permit 49...
!
router bgp 65404
 no bgp default ipv4-unicast
 neighbor ebgp-peers remote-as 65303
 address-family nsap
   neighbor ebgp-peers prefix-list routes0404 out
   neighbor 10.6.7.8 peer-group ebgp-peers
```

Example: Originating a Default Route and Outbound Route Filtering

In the figure below, autonomous system AS65101 is connected to only one other autonomous system, AS65202. Router R2 in AS65202 provides the connectivity to the rest of the network for autonomous system AS65101 by sending a default route to R1. Any packets from Level 1 routers within autonomous system AS65101 with destination NSAP addresses outside the local Level 1 network are sent to R1, the nearest Level 2 router. Router R1 forwards the packets to router R2 using the default route.

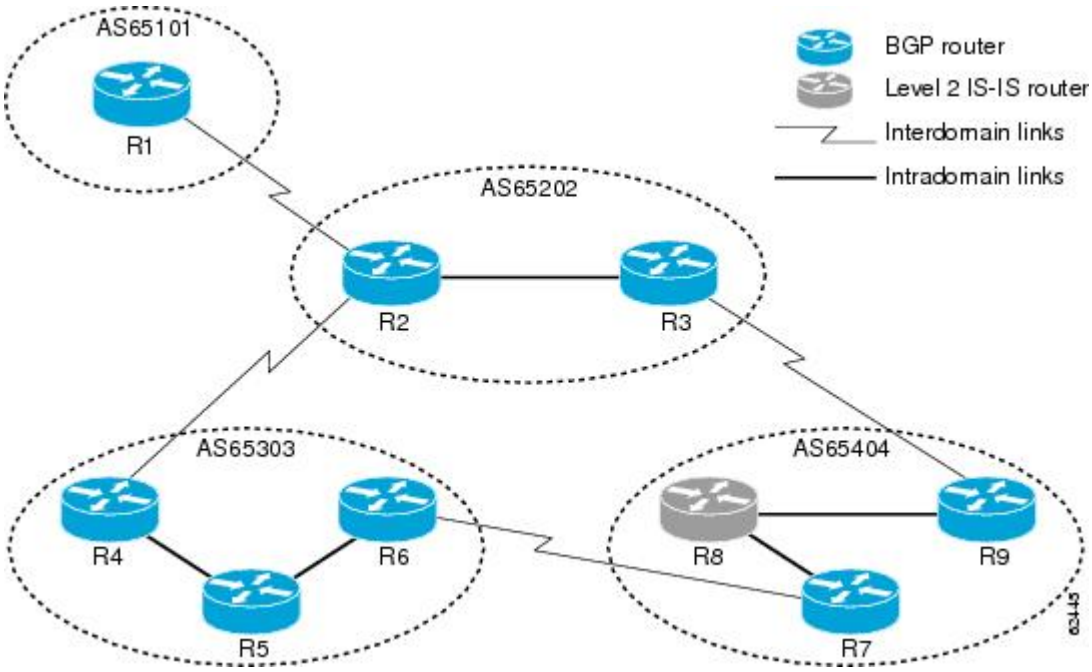
In the following example, the router R2, shown in the figure below, in the autonomous system AS65202 is configured to generate a default route for router R1 in the autonomous system AS65101, and an outbound filter is created to send only the default route NSAP addressing information in the BGP update messages to router R1.

```
clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
router bgp 65202
 no bgp default ipv4-unicast
 neighbor 10.1.2.1 remote-as 64101
 address-family nsap
  network 49.0202...
  neighbor 10.1.2.1 activate
  neighbor 10.1.2.1 default-originate
  neighbor 10.1.2.1 prefix-list default-prefix-only out
```

Implementing MP-BGP Support for CLNS Example

The figure below shows a generic BGP CLNS network containing nine routers that are grouped into four different autonomous systems (in BGP terminology) or routing domains (in OSI terminology). This section contains complete configurations for all routers shown in the figure below.

Figure 52: Components in a Generic BGP CLNS Network



If you need more details about commands used in the following examples, see the configuration tasks earlier in this document and the documents listed in the Additional References, on page 745.

Autonomous System AS65101

Router 1

```
clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
router isis osi-as-101
 net 49.0101.1111.1111.1111.1111.00
!
router bgp 65101
 no bgp default ipv4-unicast
 neighbor 10.1.2.2 remote-as 65202
 address-family nsap
  neighbor 10.1.2.2 activate
  neighbor 10.1.2.2 prefix-list default-prefix-only in
  network 49.0101...
  exit-address-family
!
interface serial 2/0
 ip address 10.1.2.1 255.255.255.0
 clns enable
 no shutdown
```

Autonomous System AS65202

Router 2

```
clns filter-set default-prefix-only deny 49...
clns filter-set default-prefix-only permit default
!
clns filter-set internal-routes permit 49.0202...
!
route-map internal-routes-only permit 10
  match clns address internal-routes
!
router isis osi-as-202
  net 49.0202.2222.2222.2222.2222.00
!
router bgp 65202
  no bgp default ipv4-unicast
  neighbor 10.1.2.1 remote-as 65101
  neighbor 10.2.3.3 remote-as 65202
  neighbor 10.2.4.4 remote-as 65303
  address-family nsap
    neighbor 10.1.2.1 activate
    neighbor 10.2.3.3 activate
    neighbor 10.2.4.4 activate
    redistribute isis osi-as-202 clns route-map internal-routes-only
    neighbor 10.1.2.1 default-originate
    neighbor 10.1.2.1 prefix-list default-prefix-only out
    exit-address-family
!
interface gigabitethernet 0/1/1
  ip address 10.2.3.2 255.255.255.0
  clns router isis osi-as-202
  no shutdown
!
interface serial 2/0
  ip address 10.1.2.2 255.255.255.0
  clns enable
  no shutdown
!
interface serial 2/2
  ip address 10.2.4.2 255.255.255.0
  clns enable
  no shutdown
```

Router 3

```
clns filter-set internal-routes permit 49.0202...
!
route-map internal-routes-only permit 10
  match clns address internal-routes
!
router isis osi-as-202
 net 49.0202.3333.3333.3333.3333.00
!
router bgp 65202
 no bgp default ipv4-unicast
 neighbor 10.2.3.2 remote-as 65202
 neighbor 10.3.9.9 remote-as 65404
 address-family nsap
  neighbor 10.2.3.2 activate
  neighbor 10.3.9.9 activate
```

```
  redistribute isis osi-as-202 clns route-map internal-routes-only
  exit-address-family
!
interface gigabitethernet 0/1/1
 ip address 10.2.3.3 255.255.255.0
 clns router isis osi-as-202
 no shutdown
!
interface serial 2/2
 ip address 10.3.9.3 255.255.255.0
 clns enable
 no shutdown
```

Autonomous System AS65303

Router 4

```
router isis osi-as-303
 net 49.0303.4444.4444.4444.4444.00
!
router bgp 65303
 no bgp default ipv4-unicast
 neighbor 10.2.4.2 remote-as 65202
 neighbor 10.4.5.5 remote-as 65303
 address-family nsap
  no synchronization
  neighbor 10.2.4.2 activate
  neighbor 10.4.5.5 activate
  network 49.0303...
  exit-address-family
!
interface gigabitethernet 0/2/1
 ip address 10.4.5.4 255.255.255.0
 clns router isis osi-as-303
 no shutdown
!
interface serial 2/3
 ip address 10.2.4.4 255.255.255.0
 clns enable
 no shutdown
```

Router 5

```
router isis osi-as-303
 net 49.0303.5555.5555.5555.5555.00
!
router bgp 65303
 no bgp default ipv4-unicast
 neighbor ibgp-peers peer-group
 neighbor ibgp-peers remote-as 65303
 address-family nsap
  no synchronization
  neighbor ibgp-peers route-reflector-client
  neighbor 10.4.5.4 peer-group ibgp-peers
  neighbor 10.5.6.6 peer-group ibgp-peers
  exit-address-family
!
interface gigabitethernet 0/2/1
 ip address 10.4.5.5 255.255.255.0
 clns router isis osi-as-303
```

```
 no shutdown
!
interface gigabitethernet 0/3/1
 ip address 10.5.6.5 255.255.255.0
 clns router isis osi-as-303
 no shutdown
```

Router 6

```
router isis osi-as-303
 net 49.0303.6666.6666.6666.6666.00
!
router bgp 65303
 no bgp default ipv4-unicast
 neighbor 10.5.6.5 remote-as 65303
 neighbor 10.6.7.7 remote-as 65404
 address-family nsap
  no synchronization
  neighbor 10.5.6.5 activate
  neighbor 10.6.7.7 activate
  network 49.0303...
!
interface gigabitethernet 0/3/1
 ip address 10.5.6.6 255.255.255.0
 clns router isis osi-as-303
 no shutdown
!
interface serial 2/2
 ip address 10.6.7.6 255.255.255.0
 clns enable
 no shutdown
```

Autonomous System AS65404

Router 7

```
clns filter-set external-routes deny 49.0404...
clns filter-set external-routes permit 49...
!
route-map noexport permit 10
  match clns address external-routes
  set community noexport
!
router isis osi-as-404
 net 49.0404.7777.7777.7777.7777.00
 redistribute bgp 404 clns
!
router bgp 65404
 no bgp default ipv4-unicast
 neighbor 10.6.7.6 remote-as 65303
 neighbor 10.8.9.9 remote-as 65404
 address-family nsap
  neighbor 10.6.7.6 activate
  neighbor 10.8.9.9 activate
  neighbor 10.8.9.9 send-community
  neighbor 10.8.9.9 route-map noexport out
  network 49.0404...
!
interface gigabitethernet 1/0/1
 ip address 10.7.8.7 255.255.255.0
```

```
 clns router isis osi-as-404
 ip router isis osi-as-404
 no shutdown
!
interface serial 2/3
 ip address 10.6.7.7 255.255.255.0
 clns enable
 no shutdown
```

Router 8

```
router isis osi-as-404
 net 49.0404.8888.8888.8888.8888.00
!
interface gigabitethernet 1/0/1
 ip address 10.7.8.8 255.255.255.0
 clns router isis osi-as-404
 ip router isis osi-as-404
 no shutdown
!
interface gigabitethernet 1/1/1
 ip address 10.8.9.8 255.255.255.0
 clns router isis osi-as-404
 ip router isis osi-as-404
 no shutdown
```

Router 9

```
clns filter-set external-routes deny 49.0404...
clns filter-set external-routes permit 49...
!
route-map noexport permit 10
  match clns address external-routes
  set community noexport
!
router isis osi-as-404
 net 49.0404.9999.9999.9999.9999.00
 redistribute bgp 404 clns
!
router bgp 65404
 no bgp default ipv4-unicast
 neighbor 10.3.9.3 remote-as 65202
 neighbor 10.7.8.7 remote-as 65404
 address-family nsap
  network 49.0404...
  neighbor 10.3.9.3 activate
  neighbor 10.7.8.7 activate
  neighbor 10.7.8.7 send-community
  neighbor 10.7.8.7 route-map noexport out
!
interface serial 2/3
 ip address 10.3.9.9 255.255.255.0
 clns enable
 no shutdown
!
interface gigabitethernet 1/1/1
 ip address 10.8.9.9 255.255.255.0
 clns router isis osi-as-404
 ip router isis osi-as-404
 no shutdown
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring MP-BGP Support for CLNS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 69: Feature Information for MP-BGP Support for CLNS

Feature Name	Releases	Feature Information
Multiprotocol BGP (MP-BGP) Support for CLNS	Cisco IOS XE Release 2.6	

Feature Name	Releases	Feature Information
		The Multiprotocol BGP (MP-BGP) Support for CLNS feature provides the ability to scale Connectionless Network Service (CLNS) networks. The multiprotocol extensions of Border Gateway Protocol (BGP) add the ability to interconnect separate Open System Interconnection (OSI) routing domains without merging the routing domains, thus providing the capability to build very large OSI networks. The following commands were introduced or modified by this feature: • **address-family nsap** • **clear bgp nsap** • **clear bgp nsap dampening** • **clear bgp nsap external** • **clear bgp nsap flap-statistics** • **clear bgp nsap peer-group** • **debug bgp nsap** • **debug bgp nsap dampening** • **debug bgp nsap updates** • **neighbor prefix-list** • **network (BGP and multiprotocol BGP)** • **redistribute (BGP to ISO ISIS)** • **redistribute (ISO ISIS to BGP)** • **show bgp nsap** • **show bgp nsap community** • **show bgp nsap community-list** • **show bgp nsap dampened-paths** • **show bgp nsap filter-list** • **show bgp nsap flap-statistics** • **show bgp nsap inconsistent-as** • **show bgp nsap neighbors** • **show bgp nsap paths** • **show bgp nsap quote-regexp** • **show bgp nsap regexp** • **show bgp nsap summary**

Glossary

address family --A group of network protocols that share a common format of network address. Address families are defined by RFC 1700.

AS --autonomous system. An IP term to describe a routing domain that has its own independent routing policy and is administered by a single authority. Equivalent to the OSI term "routing domain."

BGP --Border Gateway Protocol. Interdomain routing protocol that exchanges reachability information with other BGP systems.

CLNS --Connectionless Network Service . An OSI network-layer protocol.

CMIP --Common Management Information Protocol. In OSI, a network management protocol created and standardized by ISO for the monitoring and control of heterogeneous networks.

DCC --data communications channel.

DCN --data communications network.

ES-IS --End System-to-Intermediate System. OSI protocol that defines how end systems (hosts) announce themselves to intermediate systems (routers).

FTAM --File Transfer, Access, and Management. In OSI, an application-layer protocol developed for network file exchange and management between diverse types of computers.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within an autonomous system.

IGRP --Interior Gateway Routing Protocol. A proprietary Cisco protocol developed to address the issues associated with routing in large, heterogeneous networks.

IS --intermediate system. Routing node in an OSI network.

IS-IS --Intermediate System-to-Intermediate System. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, where routers exchange routing information based on a single metric, to determine network topology.

ISO --International Organization for Standardization. International organization that is responsible for a wide range of standards, including those relevant to networking. ISO developed the Open System Interconnection (OSI) reference model, a popular networking reference model.

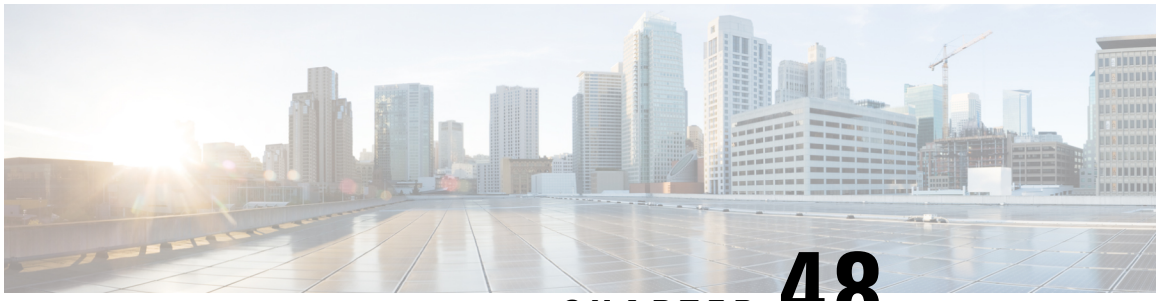
NSAP address --network service access point address. The network address format used by OSI networks.

OSI --Open System Interconnection. International standardization program created by ISO and ITU-T to develop standards for data networking that facilitate multivendor equipment interoperability.

routing domain --The OSI term that is equivalent to autonomous system for BGP.

SDH --Synchronous Digital Hierarchy. Standard that defines a set of rate and format standards that are sent using optical signals over fiber.

SONET --Synchronous Optical Network. High-speed synchronous network specification designed to run on optical fiber.



CHAPTER 48

BGP IPv6 Admin Distance

The BGP IPv6 Admin Distance feature lets you prioritize the BGP IPv6 routes in your network by enabling you to configure the source specific distance of a route and associate a prefix-list with the route. The RIB uses the distance from the source to determine the priority of the BGP IPv6 route in the network.

Information About BGP IPv6 Admin Distance

The BGP IPv6 Admin Distance feature supports selection of route path for a set prefix by prioritizing the BGP routes in the RIB. The BGP routes provided in the RIB are prioritized based on the distance they are configured from a source. With BGP IPv6 Admin Distance feature you can configure the distance from a source and can associate the route with a prefix-list. The route with the source specific distance and the prefix-list is then utilized by the RIB to prioritize the BGP IPv6 routes.

Benefits of Using BGP IPv6 Admin Distance

The BGP IPv6 Admin Distance feature can be used to prioritize or de-prioritize the BGP IPv6 routes in your network.

Configuring BGP IPv6 Admin Distance

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **router bgp***autonomous-system-number*
5. **address-family ipv6 unicast**
6. **distance** *admin-distance ipv6-address/prefix prelengthinterface nameprefix-list*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams
Step 4	**router bgp***autonomous-system-number* **Example:** `Device(config)# router bgp 5`	Specifies the number of an autonomous system that identifies the router to other BGP routers and tags the routing information that is passed along. • The range is from 1 to 65535.
Step 5	**address-family ipv6 unicast** **Example:** `Device(config-router)# address-family ipv6 unicast`	Enters the address family configuration mode for configuring routing sessions.
Step 6	**distance** *admin-distance ipv6-address/prefix prelengthinterface nameprefix-list* **Example:** `Device(config-router-af)# distance 12 2001:DB8:0:CC00::1/128 list1`	Specifies the administrative distance, IPv6 address, prefix length and prefix list name for configuring the source specific distance for BGP routes. Interface name is optional and is required only if the neighbor address is a link local address.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Verifying BGP Admin Distance Configuration

Use the **show run sec bgp** command to verify the BGP configuration:

```
Device(config-device-af)# show run | sec bgp
router bgp 200
bgp log-neighbor-changes
neighbor FE80::A8BB:CCFF:FE02:BE01%Ethernet0/0 remote-as 200
neighbor FE80::A8BB:CCFF:FE02:BE01%Ethernet0/0 update-source Ethernet0/0
```

```
!
address-family ipv4
  no neighbor FE80::A8BB:CCFF:FE02:BE01%Ethernet0/0 activate
exit-address-family
!
address-family ipv6
  distance 90 FE80::A8BB:CCFF:FE02:BE01/128 interface Ethernet0/0
  network 1:1:1:1::/120
  neighbor FE80::A8BB:CCFF:FE02:BE01%Ethernet0/0 activate
exit-address-family
```

Use the **do show ipv6 route** command to verify the IPv6 route configuration:

```
Device(config-device-af)# show ipv6 route
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
       lA - LISP away, a - Application
C   1:1:1:1::/120 [0/0]
     via Ethernet0/0, directly connected
L   1:1:1:1::2/128 [0/0]
     via Ethernet0/0, receive
B   3:4:5:6::1/128 [90/0]
     via FE80::A8BB:CCFF:FE02:BF01, Ethernet0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

Additional References for BGP IPv6 Admin Distance

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS IP Routing: BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

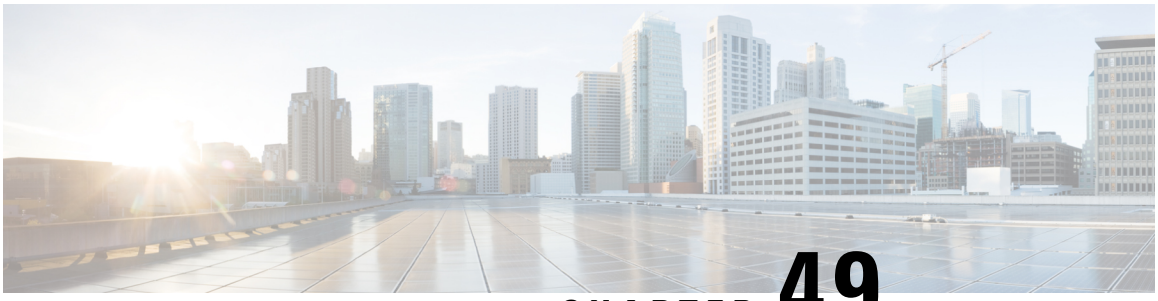
Feature Information for BGP IPv6 Admin Distance

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 70: Feature Information for ASR1K NPTv6

Feature Name	Releases	Feature Configuration Information
BGP IPv6 Admin Distance	Cisco IOS XE Denali 16.3.1	The BGP IPv6 Admin Distance feature lets you prioritize the BGP IPv6 routes in your network by enabling you to configure the source specific distance of a route and associate a prefix-list with the route. The RIB uses the distance from the source to determine the priority of the BGP IPv6 route in the network. The following commands were modified: **distance**



CHAPTER 49

Connecting to a Service Provider Using External BGP

This module describes configuration tasks that will enable your Border Gateway Protocol (BGP) network to access peer devices in external networks such as those from Internet service providers (ISPs). BGP is an interdomain routing protocol that is designed to provide loop-free routing between organizations. External BGP (eBGP) peering sessions are configured to allow peers from different autonomous systems to exchange routing updates. Tasks to help manage the traffic that is flowing inbound and outbound are described, as are tasks to configure BGP policies to filter the traffic. Multihoming techniques that provide redundancy for connections to a service provider are also described.

Prerequisites for Connecting to a Service Provider Using External BGP

- Before connecting to a service provider you need to understand how to configure the basic BGP process and peers. See the “Cisco BGP Overview” and “Configuring a Basic BGP Network” modules for more details.
- The tasks and concepts in this chapter will help you configure BGP features that would be useful if you are connecting your network to a service provider. For each connection to the Internet, you must have an assigned autonomous system number from the Internet Assigned Numbers Authority (IANA).

Restrictions for Connecting to a Service Provider Using External BGP

- A router that runs Cisco IOS software can be configured to run only one BGP routing process and to be a member of only one BGP autonomous system. However, a BGP routing process and autonomous system can support multiple address family configurations.
- Policy lists are not supported in versions of Cisco IOS software prior to Cisco IOS Release 12.0(22)S and 12.2(15)T. Reloading a router that is running an older version of Cisco IOS software may cause some routing policy configurations to be lost.

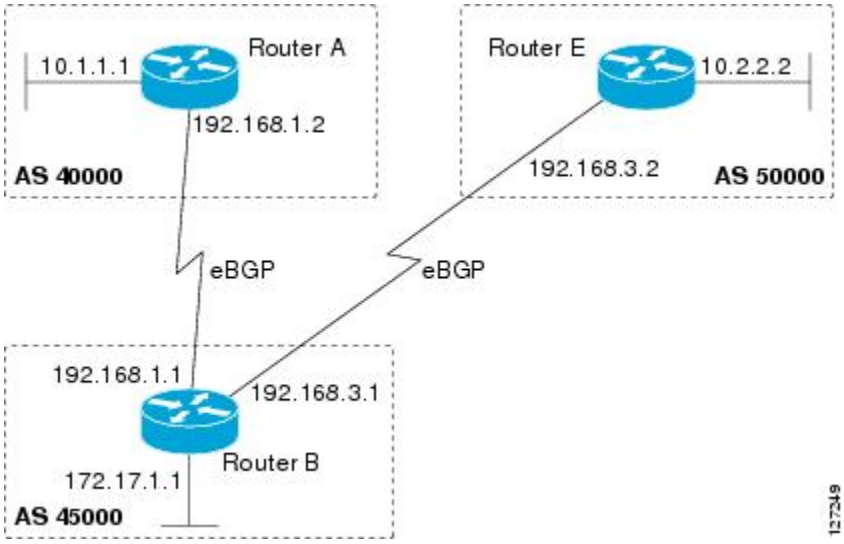
Information About Connecting to a Service Provider Using External BGP

External BGP Peering

BGP is an interdomain routing protocol designed to provide loop-free routing links between organizations. BGP is designed to run over a reliable transport protocol and it uses TCP (port 179) as the transport protocol. The destination TCP port is assigned 179, and the local port is assigned a random port number. Cisco IOS software supports BGP version 4, which has been used by ISPs to help build the Internet. RFC 1771 introduced and discussed a number of new BGP features to allow the protocol to scale for Internet use.

External BGP peering sessions are configured to allow BGP peers from different autonomous systems to exchange routing updates. By design, a BGP routing process expects eBGP peers to be directly connected, for example, over a WAN connection. However, there are many real-world scenarios where this rule would prevent routing from occurring. Peering sessions for multihop neighbors are configured with the **neighbor ebgp-multihop** command. The figure below shows simple eBGP peering between three routers. Router B peers with Router A and Router E. In the figure below, the **neighbor ebgp-multihop** command could be used to establish peering between Router A and Router E although this is a very simple network design. BGP forwards information about the next hop in the network using the NEXT_HOP attribute, which is set to the IP address of the interface that advertises a route in an eBGP peering session by default. The source interface can be a physical interface or a loopback interface.

Figure 53: BGP Peers in Different Autonomous Systems



Loopback interfaces are preferred for establishing eBGP peering sessions because loopback interfaces are less susceptible to interface flapping. Interfaces on networking devices can fail, and they can also be taken out of service for maintenance. When an interface is administratively brought up or down, due to failure or maintenance, it is referred to as a flap. Loopback interfaces provide a stable source interface to ensure that the IP address assigned to the interface is always reachable as long as the IP routing protocols continue to advertise the subnet assigned to the loopback interface. Loopback interfaces allow you to conserve address space by configuring a single address with /32 bit mask. Before a loopback interface is configured for an eBGP peering session, you must configure the **neighbor update-source** command and specify the loopback interface. With this configuration, the loopback interface becomes the source interface and its IP address is advertised as the next hop for routes that are advertised through this loopback. If loopback interfaces are used to connect single-hop eBGP peers, you must configure the **neighbor disable-connected-check** command before you can establish the eBGP peering session.

Connecting to external networks enables traffic from your network to be forwarded to other networks and across the Internet. Traffic will also be flowing into, and possibly through, your network. BGP contains various techniques to influence how the traffic flows into and out of your network, and to create BGP policies that filter the traffic, inbound and outbound. To influence the traffic flow, BGP uses certain BGP attributes that can be included in update messages or used by the BGP routing algorithm. BGP policies to filter traffic also use some of the BGP attributes with route maps, access lists including AS-path access lists, filter lists, policy lists, and distribute lists. Managing your external connections may involve multihoming techniques where there is more than one connection to an ISP or connections to more than one ISP for backup or performance purposes. Tagging BGP routes with different community attributes across autonomous system or physical boundaries can prevent the need to configure long lists of individual permit or deny statements.

BGP Autonomous System Number Formats

Prior to January 2009, BGP autonomous system numbers that were allocated to companies were 2-octet numbers in the range from 1 to 65535 as described in RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*. Due to increased demand for autonomous system numbers, the Internet Assigned Number Authority (IANA) will start in January 2009 to allocate four-octet autonomous system numbers in the range from 65536 to 4294967295. RFC 5396, *Textual Representation of Autonomous System (AS) Numbers* , documents three methods of representing autonomous system numbers. Cisco has implemented the following two methods:

- Asplain--Decimal value notation where both 2-byte and 4-byte autonomous system numbers are represented by their decimal value. For example, 65526 is a 2-byte autonomous system number and 234567 is a 4-byte autonomous system number.
- Asdot--Autonomous system dot notation where 2-byte autonomous system numbers are represented by their decimal value and 4-byte autonomous system numbers are represented by a dot notation. For example, 65526 is a 2-byte autonomous system number and 1.169031 is a 4-byte autonomous system number (this is dot notation for the 234567 decimal number).

For details about the third method of representing autonomous system numbers, see RFC 5396.

Asdot Only Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases, the 4-octet (4-byte) autonomous system numbers are entered and displayed only in asdot notation, for example, 1.10 or 45000.64000. When using regular expressions to match 4-byte autonomous system numbers the asdot format includes a period which is a special character in regular expressions. A backslash must be entered before the period for example, 1\.14, to ensure the regular expression match does not fail. The table below shows the format in which 2-byte and 4-byte autonomous system numbers are configured, matched in regular expressions, and displayed in **show** command output in Cisco IOS images where only asdot formatting is available.

Table 71: Asdot Only 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Asplain as Default Autonomous System Number Formatting

In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases, the Cisco implementation of 4-byte autonomous system numbers uses asplain as the default display format for autonomous system numbers, but you can configure 4-byte autonomous system numbers in both the asplain and asdot format. In addition, the default format for matching 4-byte autonomous system numbers in regular expressions is asplain, so you must ensure that any regular expressions to match 4-byte autonomous system numbers are written in the asplain format. If you want to change the default **show** command output to display 4-byte autonomous system numbers in the asdot format, use the **bgp asnotation dot** command under router configuration mode. When the asdot format is enabled as the default, any regular expressions to match 4-byte autonomous system numbers must be written using the asdot format, or the regular expression match will fail. The tables below show that although you can configure 4-byte autonomous system numbers in either asplain or asdot format, only one format is used to display **show** command output and control 4-byte autonomous system number matching for regular expressions, and the default is asplain format. To display 4-byte autonomous system numbers in **show** command output and to control matching for regular expressions in the asdot format, you must configure the **bgp asnotation dot** command. After enabling the **bgp asnotation dot** command, a hard reset must be initiated for all BGP sessions by entering the **clear ip bgp *** command.

Note If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The **show** command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

Table 72: Default Asplain 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 65536 to 4294967295
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 65536 to 4294967295

Table 73: Asdot 4-Byte Autonomous System Number Format

Format	Configuration Format	Show Command Output and Regular Expression Match Format
asplain	2-byte: 1 to 65535 4-byte: 65536 to 4294967295	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535
asdot	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535	2-byte: 1 to 65535 4-byte: 1.0 to 65535.65535

Reserved and Private Autonomous System Numbers

In Cisco IOS Release 12.0(32)S12, 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 12.4(24)T, and later releases, the Cisco implementation of BGP supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

RFC 5398, *Autonomous System (AS) Number Reservation for Documentation Use*, describes new reserved autonomous system numbers for documentation purposes. Use of the reserved numbers allow configuration examples to be accurately documented and avoids conflict with production networks if these configurations are literally copied. The reserved numbers are documented in the IANA autonomous system number registry. Reserved 2-byte autonomous system numbers are in the contiguous block, 64496 to 64511 and reserved 4-byte autonomous system numbers are from 65536 to 65551 inclusive.

Private 2-byte autonomous system numbers are still valid in the range from 64512 to 65534 with 65535 being reserved for special use. Private autonomous system numbers can be used for internal routing domains but must be translated for traffic that is routed out to the Internet. BGP should not be configured to advertise private autonomous system numbers to external networks. Cisco IOS software does not remove private autonomous system numbers from routing updates by default. We recommend that ISPs filter private autonomous system numbers.

Note Autonomous system number assignment for public and private networks is governed by the IANA. For information about autonomous-system numbers, including reserved number assignment, or to apply to register an autonomous system number, see the following URL: http://www.iana.org/.

BGP Attributes

BGP selects a single path, by default, as the best path to a destination host or network. The best-path selection algorithm analyzes path attributes to determine which route is installed as the best path in the BGP routing table. Each path carries various attributes that are used in BGP best-path analysis. Cisco IOS software provides the ability to influence BGP path selection by altering these attributes via the command-line interface (CLI). BGP path selection can also be influenced through standard BGP policy configuration.

BGP uses the best-path selection algorithm to find a set of equally good routes. These routes are the potential multipaths. In Cisco IOS Release 12.2(33)SRD and later releases, when there are more equally good multipaths available than the maximum permitted number, then the oldest paths are selected as multipaths.

BGP can include path attribute information in update messages. BGP attributes describe the characteristic of the route, and the software uses these attributes to help make decisions about which routes to advertise. Some of this attribute information can be configured at a BGP-speaking networking device. There are some mandatory attributes that are always included in the update message and some discretionary attributes. The following BGP attributes can be configured:

- AS_Path
- Community
- Local_Pref
- Multi_Exit_Discriminator (MED)
- Next_Hop
- Origin

AS_Path

This attribute contains a list or set of the autonomous system numbers through which routing information has passed. The BGP speaker adds its own autonomous system number to the list when it forwards the update message to external peers.

Community

BGP communities are used to group networking devices that share common properties, regardless of network, autonomous system, or any physical boundaries. In large networks applying a common routing policy through prefix lists or access lists requires individual peer statements on each networking device. Using the BGP community attribute BGP neighbors, with common routing policies, can implement inbound or outbound route filters based on the community tag rather than consult large lists of individual permit or deny statements.

Local_Pref

Within an autonomous system, the Local_Pref attribute is included in all update messages between BGP peers. If there are several paths to the same destination, the local preference attribute with the highest value indicates the preferred outbound path from the local autonomous system. The highest ranking route is advertised to internal peers. The Local_Pref value is not forwarded to external peers.

Multi_Exit_Discriminator

The MED attribute indicates (to an external peer) a preferred path into an autonomous system. If there are multiple entry points into an autonomous system, the MED can be used to influence another autonomous system to choose one particular entry point. A metric is assigned where a lower MED metric is preferred by the software over a higher MED metric. The MED metric is exchanged between autonomous systems, but after a MED is forwarded into an autonomous system, the MED metric is reset to the default value of 0. When an update is sent to an internal BGP (iBGP) peer, the MED is passed along without any change, allowing all the peers in the same autonomous system to make a consistent path selection.

By default, a router will compare the MED attribute for paths only from BGP peers that reside in the same autonomous system. The **bgp always-compare-med** command can be configured to allow the router to compare metrics from peers in different autonomous systems.

Note The Internet Engineering Task Force (IETF) decision regarding BGP MED assigns a value of infinity to the missing MED, making the route that lacks the MED variable the least preferred. The default behavior of BGP routers that run Cisco software is to treat routes without the MED attribute as having a MED of 0, making the route that lacks the MED variable the most preferred. To configure the router to conform to the IETF standard, use the **bgp bestpath med missing-as-worst** router configuration command.

Next_Hop

The Next_Hop attribute identifies the next-hop IP address to be used as the BGP next hop to the destination. The router makes a recursive lookup to find the BGP next hop in the routing table. In external BGP (eBGP), the next hop is the IP address of the peer that sent the update. Internal BGP (iBGP) sets the next-hop address to the IP address of the peer that advertised the prefix for routes that originate internally. When any routes to iBGP that are learned from eBGP are advertised, the Next_Hop attribute is unchanged.

A BGP next-hop IP address must be reachable in order for the router to use a BGP route. Reachability information is usually provided by the IGP, and changes in the IGP can influence the forwarding of the next-hop address over a network backbone.

Origin

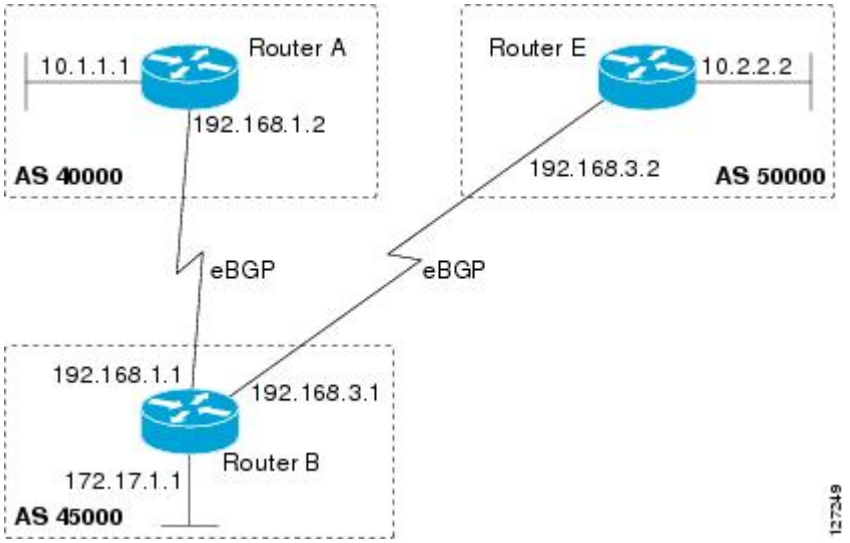
This attribute indicates how the route was included in a BGP routing table. In Cisco software, a route defined using the BGP **network** command is given an origin code of Interior Gateway Protocol (IGP). Routes distributed from an Exterior Gateway Protocol (EGP) are coded with an origin of EGP, and routes redistributed from other protocols are defined as Incomplete. BGP decision policy for origin prefers IGP over EGP, and then EGP over Incomplete.

Multihoming

Multihoming is defined as connecting an autonomous system with more than one service provider. If you have any reliability issues with one service provider, then you have a backup connection. Performance issues can also be addressed by multihoming because better paths to the destination network can be utilized.

Unless you are a service provider, you must plan your routing configuration carefully to avoid Internet traffic traveling through your autonomous system and consuming all your bandwidth. The figure below shows that autonomous system 45000 is multihomed to autonomous system 40000 and autonomous system 50000. Assuming autonomous system 45000 is not a service provider, then several techniques such as load balancing or some form of routing policy must be configured to allow traffic from autonomous system 45000 to reach either autonomous system 40000 or autonomous system 50000 but not allow much, if any, transit traffic.

Figure 54: Multihoming Topology



MED Attribute

Configuring the MED attribute is another method that BGP can use to influence the choice of paths into another autonomous system. The MED attribute indicates (to an external peer) a preferred path into an autonomous system. If there are multiple entry points into an autonomous system, the MED can be used to influence another autonomous system to choose one particular entry point. A metric is assigned using route maps where a lower MED metric is preferred by the software over a higher MED metric.

Transit Versus Nontransit Traffic

Most of the traffic within an autonomous system contains a source or destination IP address residing within the autonomous system, and this traffic is referred to as nontransit (or local) traffic. Other traffic is defined as transit traffic. As traffic across the Internet increases, controlling transit traffic becomes more important.

A service provider is considered to be a transit autonomous system and must provide connectivity to all other transit providers. In reality, few service providers actually have enough bandwidth to allow all transit traffic, and most service providers have to purchase such connectivity from Tier 1 service providers.

An autonomous system that does not usually allow transit traffic is called a stub autonomous system and will link to the Internet through one service provider.

BGP Policy Configuration

BGP policy configuration is used to control prefix processing by the BGP routing process and to filter routes from inbound and outbound advertisements. Prefix processing can be controlled by adjusting BGP timers, altering how BGP handles path attributes, limiting the number of prefixes that the routing process will accept, and configuring BGP prefix dampening. Prefixes in inbound and outbound advertisements are filtered using route maps, filter lists, IP prefix lists, autonomous-system-path access lists, IP policy lists, and distribute lists. The table below shows the processing order of BGP policy filters.

Table 74: BGP Policy Processing Order

Inbound	Outbound
Route map	Distribute list
Filter list, AS-path access list, or IP policy	IP prefix list
IP prefix list	Filter list, AS-path access list, or IP policy
Distribute list	Route map

Note In Cisco IOS Releases 12.0(22)S, 12.2(15)T, 12.2(18)S, and later releases, the maximum number of autonomous system access lists that can be configured with the **ip as-path access-list** command is increased from 199 to 500.

Whenever there is a change in the routing policy due to a configuration change, BGP peering sessions must be reset using the **clear ip bgp** command. Cisco IOS software supports the following three mechanisms to reset BGP peering sessions:

- Hard reset--A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer.
- Soft reset--A soft reset uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions. Soft reset uses stored update information, at the cost of additional memory for storing the updates, to allow you to apply a new BGP policy without disrupting the network. Soft reset can be configured for inbound or outbound sessions.
- Dynamic inbound soft reset--The route refresh capability, as defined in RFC 2918, allows the local router to reset inbound routing tables dynamically by exchanging route refresh requests to supporting peers. The route refresh capability does not store update information locally for nondisruptive policy changes. It instead relies on dynamic exchange with supporting peers. Route refresh must first be advertised through BGP capability negotiation between peers. All BGP routers must support the route refresh capability.

To determine if a BGP router supports this capability, use the **show ip bgp neighbors**command. The following message is displayed in the output when the router supports the route refresh capability:

```
Received route refresh capability from peer.
```

BGP COMMUNITIES Attribute

A BGP community is a group of routes that share a common property, regardless of their network, autonomous system, or any physical boundaries. In large networks, applying a common routing policy by using prefix lists or access lists requires individual peer statements on each networking device. Using the BGP COMMUNITIES attribute, BGP speakers with common routing policies can implement inbound or outbound route filters based on the community tag, rather than consult long lists of individual permit or deny statements. A COMMUNITIES attribute can contain multiple communities.

A route can belong to multiple communities. The network administrator defines the communities to which a route belongs. By default, all routes belong to the general Internet community.

In addition to numbered communities, there are several predefined (well-known) communities:

- no-export—Do not advertise this route to external BGP peers.
- no-advertise—Do not advertise this route to any peer.
- internet—Advertise this route to the Internet community. All BGP-speaking networking devices belong to this community.
- local-as—Do not send this route outside the local autonomous system.
- gshut—Community of routes gracefully shut down.

The COMMUNITIES attribute is optional, which means that it will not be passed on by networking devices that do not understand communities. Networking devices that understand communities must be configured to handle the communities or else the COMMUNITIES attribute will be discarded. By default, no COMMUNITIES attribute is sent to a neighbor. In order for a COMMUNITIES attribute to be sent to a neighbor, use the **neighbor send-community** command.

Extended Communities

Extended community attributes are used to configure, filter, and identify routes for virtual routing and forwarding (VRF) instances and Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). All of the standard rules of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers. All regular expression configuration options are supported. The route target (RT) and site of origin (SoO) extended community attributes are supported by the standard range of extended community lists.

Route Target Extended Community Attribute

The RT extended community attribute is configured with the **rt** keyword of the **ip extcommunity-list** command. This attribute is used to identify a set of sites and VRFs that may receive routes that are tagged with the configured route target. Configuring the route target extended community attribute with a route allows that route to be placed in the per-site forwarding tables that are used for routing traffic that is received from corresponding sites.

Site of Origin Extended Community Attribute

The SoO extended community attribute is configured with the **soo** keyword of the **ip extcommunity-list** command. This attribute uniquely identifies the site from which the provider edge (PE) router learned the route. All routes learned from a particular site must be assigned the same SoO extended community attribute, regardless if a site is connected to a single PE router or multiple PE routers. Configuring this attribute prevents

routing loops from occurring when a site is multihomed. The SoO extended community attribute is configured on the interface and is propagated into BGP through redistribution. The SoO extended community attribute can be applied to routes that are learned from VRFs. The SoO extended community attribute should not be configured for stub sites or sites that are not multihomed.

IP Extended Community-List Configuration Mode

Named and numbered extended community lists can be configured in IP extended community-list configuration mode. The IP extended community-list configuration mode supports all of the functions that are available in global configuration mode. In addition, the following operations can be performed:

- Configure sequence numbers for extended community list entries.
- Resequence existing sequence numbers for extended community list entries.
- Configure an extended community list to use default values.

Default Sequence Numbering

Extended community list entries start with the number 10 and increment by 10 for each subsequent entry when no sequence number is specified, when default behavior is configured, and when an extended community list is resequenced without specifying the first entry number or the increment range for subsequent entries.

Resequencing Extended Community Lists

Extended community-list entries are sequenced and resequenced on a per-extended community list basis. The **resequence** command can be used without any arguments to set all entries in a list to default sequence numbering. The **resequence** command also allows the sequence number of the first entry and increment range to be set for each subsequent entry. The range of configurable sequence numbers is from 1 to 2147483647.

Extended Community Lists

Extended community attributes are used to configure, filter, and identify routes for VRF instances and MPLS VPNs. The **ip extcommunity-list** command is used to configure named or numbered extended community lists. All of the standard rules of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers.

Administrative Distance

Administrative distance is a measure of the preference of different routing protocols. BGP has a **distance bgp** command that allows you to set different administrative distances for three route types: external, internal, and local. BGP, like other protocols, prefers the route with the lowest administrative distance.

BGP Route Map Policy Lists

BGP route map policy lists allow a network operator to group route map match clauses into named lists called policy lists. A policy list functions like a macro. When a policy list is referenced in a route map, all of the match clauses are evaluated and processed as if they had been configured directly in the route map. This enhancement simplifies the configuration of BGP routing policy in medium-size and large networks because a network operator can preconfigure policy lists with groups of match clauses and then reference these policy

lists within different route maps. The network operator no longer needs to manually reconfigure each recurring group of match clauses that occur in multiple route map entries.

A policy lists functions like a macro when it is configured in a route map and has the following capabilities and characteristics:

- When a policy list is referenced within a route map, all the match statements within the policy list are evaluated and processed.
- Two or more policy lists can be configured with a route map. Policy lists can be configured within a route map to be evaluated with AND or OR semantics.
- Policy lists can coexist with any other preexisting match and set statements that are configured within the same route map but outside of the policy lists.
- When multiple policy lists perform matching within a route map entry, all policy lists match on the incoming attribute only.

Policy lists support only match clauses and do not support set clauses. Policy lists can be configured for all applications of route maps, including redistribution, and can also coexist, within the same route map entry, with match and set clauses that are configured separately from the policy lists.

Note Policy lists are supported only by BGP and are not supported by other IP routing protocols.

EBGP Route Propagation without Policies

By default, an External BGP (EBGP) router propagates routes to and from an EBGP neighbor when you have not configured inbound and outbound policies . From Cisco IOS XE Release 17.2.1, you can modify this default behavior. You can configure an EBGP router not to propagate routes to and from a neighbor unless you configure at least one inbound and one outbound policy for the neighbor.

Benefits

By preventing an EBGP router from propagating routes when inbound and outbound policies aren't configured, you prevent security issues, and business and technical impacts due to:

- route leaks that could cause traffic to be routed through unexpected paths
- software defects and misconfiguration

Restrictions for EBGP Route Propagation without Policies

- When you configure **bgp safe-ebgp-policy**, if inbound and outbound policies are not configured, an EBGP router does not propagate routes to and from real EBGP peers. This feature does not affect peers between sub-Autonomous Systems (sub-ASs) in a confederation of Autonomous Systems (ASs).
- When you configure **bgp safe-ebgp-policy**, if inbound and outbound policies are not configured, an EBGP router does not start propagating routes when it receives an outbound policy from a peer through ORF or RTC.
- *RFC 8212 - Default External BGP (EBGP) Route Propagation Behavior without Policies* recommends that routes should be omitted from Adj-RIB-In if an inbound policy is not configured and from

Adj-RIB-Out if an outbound policy is not configured. When you configure **bgp safe-ebgp-policy**, routes are omitted from Adj-RIB-In and Adj-RIB-Out unless at least an inbound policy and an outbound policy are both configured.

Usage Notes for EBGP Route Propagation without Policies

- You can prevent an EBGP router from propagating routes in the absence of any inbound and outbound policies using the **bgp safe-ebgp-policy** configuration option.
- The **bgp safe-ebgp-policy** configuration option applies globally to sessions with all EBGP neighbors but is evaluated for each Address Family (AF).

  Suppose, for a neighbor, you configure inbound and outbound policies for the IPv4 AF but not the IPv6 AF. The router accepts and advertises IPv4 routes but not IPv6 routes.
- After you configure **bgp safe-ebgp-policy**, the router propagates routes to and from a neighbor only if you configure at least one inbound and one outbound policy.
- To configure an inbound policy for a neighbor, configure at least one of the following:
  - prefix-list inbound
  - distribute-list inbound
  - filter-list inbound
  - route-map inbound
- To configure an outbound policy for a neighbor, configure at least one of the following:
  - prefix-list outbound
  - distribute-list outbound
  - filter-list outbound
  - route-map outbound
  - unsupress-map
- When you configure **bgp safe-ebgp-policy**, inbound and outbound refresh are triggered for all neighbors for which you haven't configured at least one inbound and one outbound policy configured.

  Similarly, if you remove the last inbound or outbound policy, or configure the first inbound and outbound policy for a neighbor, inbound and outbound refresh are triggered for the neighbor.

Note If route refresh is not negotiated, schedule a hard reset or trigger a soft refresh outbound at the peering BGP speaker.

- When you remove **bgp safe-ebgp-policy**, inbound and outbound refresh are triggered for all neighbors that do not have at least one inbound and one outbound policy configured. The default behavior of propagating routes to and from neighbors when inbound and outbound policies are not configured is restored.

Note If route refresh is not negotiated, schedule a hard reset or trigger a soft refresh outbound at the peering BGP speaker.

- If you configure **bgp safe-ebgp-policy** along with **soft-reconfiguration inbound**, and do not configure at least one inbound and outbound policy, routes are received and marked as 'received-only'. However, these routes are not considered for best-path computation.

How to Connect to a Service Provider Using External BGP

Influencing Inbound Path Selection

BGP can be used to influence the choice of paths in another autonomous system. There may be several reasons for wanting BGP to choose a path that is not the obvious best route, for example, to avoid some types of transit traffic passing through an autonomous system or perhaps to avoid a very slow or congested link. BGP can influence inbound path selection using one of the following BGP attributes:

- AS-path
- Multi-Exit Discriminator (MED)

Influencing Inbound Path Selection by Modifying the AS_PATH Attribute

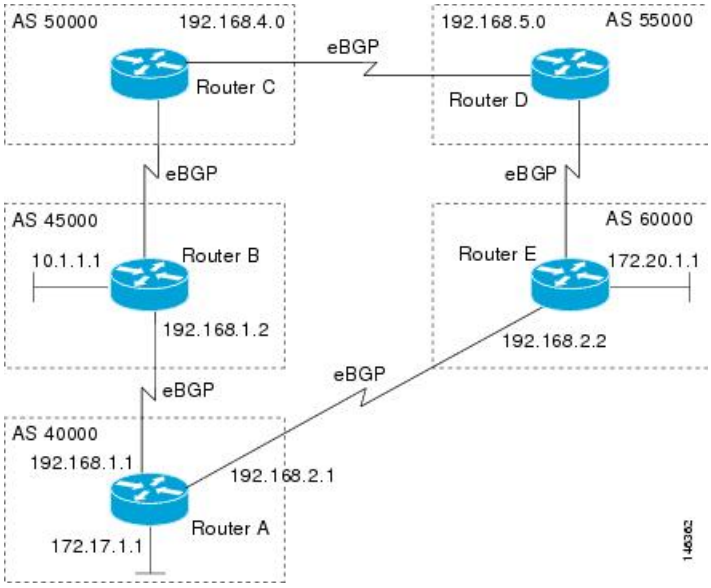
Perform this task to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS_PATH attribute. The configuration is performed at Router A in the figure below. For a configuration example of this task using 4-byte autonomous system numbers in asplain format, see the "Example: Influencing Inbound Path Selection by Modifying the AS_PATH Attribute Using 4-Byte AS Numbers".

One of the methods that BGP can use to influence the choice of paths in another autonomous system is to modify the AS_PATH attribute. For example, in the figure below, Router A advertises its own network, 172.17.1.0, to its BGP peers in autonomous system 45000 and autonomous system 60000. When the routing information is propagated to autonomous system 50000, the routers in autonomous system 50000 have network reachability information about network 172.17.1.0 from two different routes. The first route is from autonomous system 45000 with an AS_PATH consisting of 45000, 40000, the second route is through autonomous system 55000 with an AS-path of 55000, 60000, 40000. If all other BGP attribute values are the same, Router C in autonomous system 50000 would choose the route through autonomous system 45000 for traffic destined for network 172.17.1.0 because it is the shortest route in terms of autonomous systems traversed.

Autonomous system 40000 now receives all traffic from autonomous system 50000 for the 172.17.1.0 network through autonomous system 45000. If, however, the link between autonomous system 45000 and autonomous system 40000 is a really slow and congested link, the **set as-path prepend** command can be used at Router A to influence inbound path selection for the 172.17.1.0 network by making the route through autonomous system 45000 appear to be longer than the path through autonomous system 60000. The configuration is done at Router A in the figure below by applying a route map to the outbound BGP updates to Router B. Using the **set as-path prepend** command, all the outbound BGP updates from Router A to Router B will have their AS_PATH attribute modified to add the local autonomous system number 40000 twice. After the configuration, autonomous system 50000 receives updates about the 172.17.1.0 network through autonomous system 45000.

The new AS_PATH is 45000, 40000, 40000, and 40000, which is now longer than the AS-path from autonomous system 55000 (unchanged at a value of 55000, 60000, 40000). Networking devices in autonomous system 50000 will now prefer the route through autonomous system 55000 to forward packets with a destination address in the 172.17.1.0 network.

Figure 55: Network Topology for Modifying the AS_PATH Attribute



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
9. **exit-address-family**
10. **exit**
11. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
12. **set as-path** {**tag** | **prepend** *as-path-string*}
13. **end**
14. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 45000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • In this example, the BGP peer on Router B at 192.168.1.2 is added to the IPv4 multiprotocol BGP neighbor table and will receive BGP updates.
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The unicast keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.2 activate`	Enables address exchange for address family IPv4 unicast for the BGP neighbor at 192.168.1.2 on Router B.

	Command or Action	Purpose
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 192.168.1.2 route-map PREPEND out`	Applies a route map to incoming or outgoing routes. • In this example, the route map named PREPEND is applied to outbound routes to Router B.
Step 9	**exit-address-family** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 10	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 11	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map PREPEND permit 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named PREPEND is created with a permit clause.
Step 12	**set as-path** {**tag** | **prepend** *as-path-string*} **Example:** `Device(config-route-map)# set as-path prepend 40000 40000`	Modifies an autonomous system path for BGP routes. • Use the **prepend** keyword to prepend an arbitrary autonomous system path string to BGP routes. Usually the local autonomous system number is prepended multiple times, increasing the autonomous system path length. • In this example, two additional autonomous system entries are added to the autonomous system path for outbound routes to Router B.
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route map configuration mode and returns to privileged EXEC mode.
Step 14	**show running-config** **Example:** `Device# show running-config`	Displays the running configuration file.

Examples

Router A

The following partial output of the **show running-config** command shows the configuration from this task.

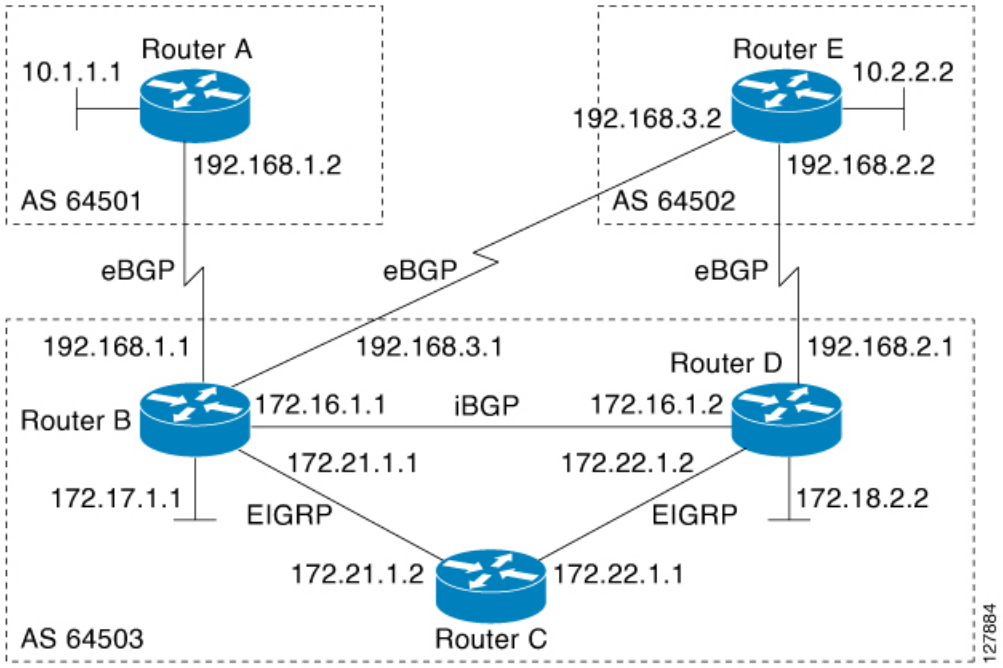
```
Device# show running-config
.
.
.
router bgp 40000
 neighbor 192.168.1.2 remote-as 45000
 !
 address-family ipv4
 neighbor 192.168.1.2 activate
 neighbor 192.168.1.2 route-map PREPEND out
 no auto-summary
 no synchronization
 network 172.17.1.0 mask 255.255.255.0
 exit-address-family
!
route-map PREPEND permit 10
 set as-path prepend 40000 40000
.
.
.
```

Influencing Inbound Path Selection by Setting the MED Attribute

One of the methods that BGP can use to influence the choice of paths into another autonomous system is to set the Multi-Exit Discriminator (MED) attribute. The MED attribute indicates (to an external peer) a preferred path to an autonomous system. If there are multiple entry points to an autonomous system, the MED can be used to influence another autonomous system to choose one particular entry point. A metric is assigned using route maps where a lower MED metric is preferred by the software over a higher MED metric.

Perform this task to influence inbound path selection by setting the MED metric attribute. The configuration is performed at Router B and Router D in the figure below. Router B advertises the network 172.16.1.0. to its BGP peer, Router E in autonomous system 50000. Using a simple route map Router B sets the MED metric to 50 for outbound updates. The task is repeated at Router D but the MED metric is set to 120. When Router E receives the updates from both Router B and Router D the MED metric is stored in the BGP routing table. Before forwarding packets to network 172.16.1.0, Router E compares the attributes from peers in the same autonomous system (both Router B and Router D are in autonomous system 45000). The MED metric for Router B is less than the MED for Router D, so Router E will forward the packets through Router B.

Figure 56: Network Topology for Setting the MED Attribute



Use the **bgp always-compare-med** command to compare MED attributes from peers in other autonomous systems.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. **exit**
10. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
11. **set metric** *value*
12. **end**
13. Repeat Step 1 through Step 12 at Router D.
14. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.3.2 remote-as 50000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Device(config-router-af)# network 172.16.1.0 mask 255.255.255.0`	Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:**	Applies a route map to incoming or outgoing routes. • In this example, the route map named MED is applied to outbound routes to the BGP peer at Router E.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.3.2 route-map MED out`	
Step 8	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 9	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 10	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map MED permit 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named MED is created.
Step 11	**set metric** *value* **Example:** `Device(config-route-map)# set metric 50`	Sets the MED metric value.
Step 12	**end** **Example:** `Device(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.
Step 13	Repeat Step 1 through Step 12 at Router D.	—
Step 14	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp 172.17.1.0 255.255.255.0`	(Optional) Displays the entries in the BGP routing table. • Use this command at Router E in the figure above when both Router B and Router D have configured the MED attribute. • Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following output is from Router E in the figure above after this task has been performed at both Router B and Router D. Note the metric (MED) values for the two routes to network 172.16.1.0. The peer 192.168.2.1 at Router D has a metric of 120 for the path to network 172.16.1.0, whereas the peer 192.168.3.1 at Router B has a metric of 50. The entry for the peer 192.168.3.1 at Router B has

the word best at the end of the entry to show that Router E will choose to send packets destined for network 172.16.1.0 via Router B because the MED metric is lower.

```
Device# show ip bgp 172.16.1.0

BGP routing table entry for 172.16.1.0/24, version 10
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  45000
    192.168.2.1 from 192.168.2.1 (192.168.2.1)
      Origin IGP, metric 120, localpref 100, valid, external
  45000
    192.168.3.1 from 192.168.3.1 (172.17.1.99)
      Origin IGP, metric 50, localpref 100, valid, external, best
```

Influencing Outbound Path Selection

BGP can be used to influence the choice of paths for outbound traffic from the local autonomous system. This section contains two methods that BGP can use to influence outbound path selection:

- Using the Local_Pref attribute
- Using the BGP outbound route filter (ORF) capability

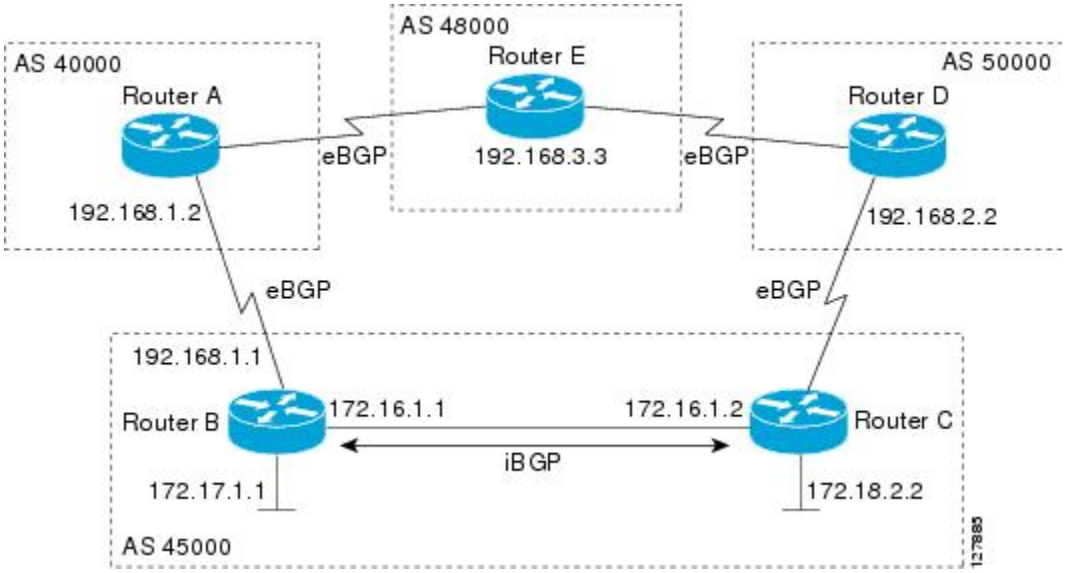
Perform one of the following tasks to influence outbound path selection:

Influencing Outbound Path Selection Using the Local_Pref Attribute

One of the methods to influence outbound path selection is to use the BGP Local-Pref attribute. Perform this task using the local preference attribute to influence outbound path selection. If there are several paths to the same destination the local preference attribute with the highest value indicates the preferred path.

Refer to the figure below for the network topology used in this task. Both Router B and Router C are configured. autonomous system 45000 receives updates for network 192.168.3.0 via autonomous system 40000 and autonomous system 50000. Router B is configured to set the local preference value to 150 for all updates to autonomous system 40000. Router C is configured to set the local preference value for all updates to autonomous system 50000 to 200. After the configuration, local preference information is exchanged within autonomous system 45000. Router B and Router C now see that updates for network 192.168.3.0 have a higher preference value from autonomous system 50000 so all traffic in autonomous system 45000 with a destination network of 192.168.3.0 is sent out via Router C.

Figure 57: Network Topology for Outbound Path Selection



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
5. **bgp default local-preference** *value*
6. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*][**route-map** *route-map-name*]
8. **neighbor** {*ip-address*| *peer-group-name*} **activate**
9. **end**
10. Repeat Step 1 through Step 9 at Router C but change the IP address of the peer, the autonomous system number, and set the local preference value to 200.
11. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**bgp default local-preference** *value* **Example:** `Router(config-router)# bgp default local-preference 150`	Changes the default local preference value. • In this example, the local preference is changed to 150 for all updates from autonomous system 40000 to autonomous system 45000. • By default, the local preference value is 100.
Step 6	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	**network** *network-number* [**mask** *network-mask*][**route-map** *route-map-name*] **Example:** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 8	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.2 activate`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

	Command or Action	Purpose
Step 9	**end** **Example:** `Router(config-router-af)# end`	Exits route map configuration mode and enters privileged EXEC mode.
Step 10	Repeat Step 1 through Step 9 at Router C but change the IP address of the peer, the autonomous system number, and set the local preference value to 200.	--
Step 11	**show ip bgp** [*network*] [*network-mask*] **Example:** `Router# show ip bgp 192.168.3.0 255.255.255.0`	Displays the entries in the BGP routing table. • Enter this command at both Router B and Router C and note the Local_Pref value. The route with the highest preference value will be the preferred route to network 192.168.3.0. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Filtering Outbound BGP Route Prefixes

Perform this task to use BGP prefix-based outbound route filtering to influence outbound path selection.

Before you begin

BGP peering sessions must be established, and BGP ORF capabilities must be enabled on each participating router before prefix-based ORF announcements can be received.

Note
- BGP prefix-based outbound route filtering does not support multicast.
- IP addresses that are used for outbound route filtering must be defined in an IP prefix list. BGP distribute lists and IP access lists are not supported.
- Outbound route filtering is configured on only a per-address family basis and cannot be configured under the general session or BGP routing process.
- Outbound route filtering is configured for external peering sessions only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
4. **router bgp** autonomous-system-*number*
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **ebgp-multihop** [*hop-count*]
7. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]

8. **neighbor** *ip-address* **capability orf prefix-list** [**send** | **receive** | **both**]
9. **neighbor** {*ip-address* | *peer-group-name*} **prefix-list** *prefix-list-name* {**in** | **out**}
10. **end**
11. **clear ip bgp** {*ip-address* | *****} **in prefix-filter**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list FILTER seq 10 permit 192.168.1.0/24`	Creates a prefix list for prefix-based outbound route filtering. • Outbound route filtering supports prefix length matching, wildcard-based prefix matching, and exact address prefix matching on a per address-family basis. • The prefix list is created to define the outbound route filter. The filter must be created when the outbound route filtering capability is configured to be advertised in send mode or both mode. It is not required when a peer is configured to advertise receive mode only. • The example creates a prefix list named FILTER that defines the 192.168.1.0/24 subnet for outbound route filtering.
Step 4	**router bgp** autonomous-system-*number* **Example:** `Router(config)# router bgp 100`	Enters router configuration mode, and creates a BGP routing process.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.1.1.1 remote-as 200`	Establishes peering with the specified neighbor or peer group. BGP peering must be established before ORF capabilities can be exchanged. • The example establishes peering with the 10.1.1.1 neighbor.
Step 6	**neighbor** *ip-address* **ebgp-multihop** [*hop-count*] **Example:**	Accepts or initiates BGP connections to external peers residing on networks that are not directly connected.

	Command or Action	Purpose
	`Router(config-router)# neighbor 10.1.1.1 ebgp-multihop`	
Step 7	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands. **Note** Outbound route filtering is configured on a per-address family basis.
Step 8	**neighbor** *ip-address* **capability orf prefix-list** [**send** | **receive** | **both**] **Example:** `Router(config-router-af)# neighbor 10.1.1.1 capability orf prefix-list both`	Enables the ORF capability on the local router, and enables ORF capability advertisement to the BGP peer specified with the *ip-address* argument. • The **send** keyword configures a router to advertise ORF send capabilities. • The **receive** keyword configures a router to advertise ORF receive capabilities. • The **both** keyword configures a router to advertise send and receive capabilities. • The remote peer must be configured to either send or receive ORF capabilities before outbound route filtering is enabled. • The example configures the router to advertise send and receive capabilities to the 10.1.1.1 neighbor.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **prefix-list** *prefix-list-name* {**in** | **out**} **Example:** `Router(config-router-af)# neighbor 10.1.1.1 prefix-list FILTER in`	Applies an inbound prefix-list filter to prevent distribution of BGP neighbor information. • In this example, the prefix list named FILTER is applied to incoming advertisements from the 10.1.1.1 neighbor, which prevents distribution of the 192.168.1.0/24 subnet.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode, and enters privileged EXEC mode.
Step 11	**clear ip bgp** {*ip-address* | *****} **in prefix-filter** **Example:** `Router# clear ip bgp 10.1.1.1 in prefix-filter`	Clears BGP outbound route filters and initiates an inbound soft reset. • A single neighbor or all neighbors can be specified. **Note** The inbound soft refresh must be initiated with the **clear ip bgp** command in order for this feature to function.

Configuring BGP Peering with ISPs

BGP was developed as an interdomain routing protocol and connecting to ISPs is one of the main functions of BGP. Depending on the size of your network and the purpose of your business, there are many different ways to connect to your ISP. Multihoming to one or more ISPs provides redundancy in case an external link to an ISP fails. This section introduces some optional tasks that can be used to connect to a service provider using multihoming techniques. Smaller companies may use just one ISP but require a backup route to the ISP. Larger companies may have access to two ISPs, using one of the connections as a backup, or may need to configure a transit autonomous system.

Perform one of the following optional tasks to connect to one or more ISPs:

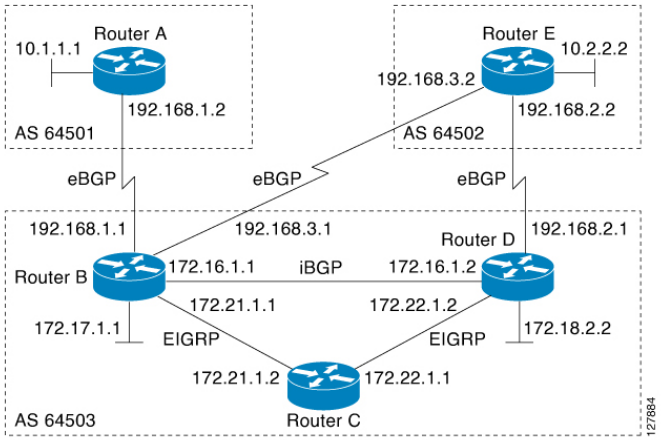
Configuring Multihoming with Two ISPs

Perform this task to configure your network to access two ISPs where one ISP is the preferred route and the second ISP is a backup route. In the figure below Router B in autonomous system 45000 has BGP peers in two ISPs, autonomous system 40000 and autonomous system 50000. Using this task, Router B will be configured to prefer the route to the BGP peer at Router A in autonomous system 40000.

All routes learned from this neighbor will have an assigned weight. The route with the highest weight will be chosen as the preferred route when multiple routes are available to a particular network.

Note The weights assigned with the **set weight** route-map configuration command override the weights assigned using the **neighbor weight** command.

Figure 58: Multihoming with Two ISPs



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* [**mask** *network-mask*]
7. **neighbor** {*ip-address* | *peer-group-name*} **weight** *number*
8. **exit**
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *peer-group-name*} **weight** *number*
12. **end**
13. **clear ip bgp** {***** | *ip-address* | *peer-group-name*} [**soft** [**in** | **out**]]
14. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *network-number* [**mask** *network-mask*] **Example:** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **weight** *number* **Example:** `Router(config-router-af)# neighbor 192.168.1.2 weight 150`	Assigns a weight to a BGP peer connection. • In this example, the weight attribute for routes received from the BGP peer 192.168.1.2 is set to 150.
Step 8	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.

	Command or Action	Purpose
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 10	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **weight** *number* **Example:** `Router(config-router-af)# neighbor 192.168.3.2 weight 100`	Assigns a weight to a BGP peer connection. • In this example, the weight attribute for routes received from the BGP peer 192.168.3.2 is set to 100.
Step 12	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 13	**clear ip bgp** {***** | *ip-address* | *peer-group-name*} [**soft** [**in** | **out**]] **Example:** `Router# clear ip bgp *`	(Optional) Clears BGP outbound route filters and initiates an outbound soft reset. A single neighbor or all neighbors can be specified.
Step 14	**show ip bgp** [*network*] [*network-mask*] **Example:** `Router# show ip bgp`	Displays the entries in the BGP routing table. • Enter this command at Router B to see the weight attribute for each route to a BGP peer. The route with the highest weight attribute will be the preferred route to network 172.17.1.0. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following example shows the BGP routing table at Router B with the weight attributes assigned to routes. The route through 192.168.1.2 (Router A in the figure above) has the highest weight attribute and will be the preferred route to network 10.3.0.0, wherein the network 10.3.0.0 is accessible through Router A and Router E. If this route (through Router B) fails for some reason, the route through 192.168.3.2 (Router E) will be used to reach network 10.3.0.0. This way, redundancy is provided for reaching Router B.

```
BGP table version is 8, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric    LocPrf   Weight  Path

*> 10.1.1.0/24         192.168.1.2           0             150      40000 i
*> 10.2.2.0/24         192.168.3.2           0             100      50000 i

*> 10.3.0.0/16         192.168.1.2           0             150      40000 i
*                      192.168.3.2           0             100      50000 i

*> 172.17.1.0/24       0.0.0.0               0             32768          i
```

Multihoming with a Single ISP

Perform this task to configure your network to access one of two connections to a single ISP, where one of the connections is the preferred route and the second connection is a backup route. In the figure above Router E in autonomous system 50000 has two BGP peers in a single autonomous system, autonomous system 45000. Using this task, autonomous system 50000 does not learn any routes from autonomous system 45000 and is sending its own routes using BGP. This task is configured at Router E in the figure above and covers three features about multihoming to a single ISP:

- Outbound traffic—Router E will forward default routes and traffic to autonomous system 45000 with Router B as the primary link and Router D as the backup link. Static routes are configured to both Router B and Router D with a lower distance configured for the link to Router B.
- Inbound traffic—Inbound traffic from autonomous system 45000 is configured to be sent from Router B unless the link fails when the backup route is to send traffic from Router D. To achieve this, outbound filters are set using the MED metric.
- Prevention of transit traffic—A route map is configured at Router E in autonomous system 50000 to block all incoming BGP routing updates to prevent autonomous system 50000 from receiving transit traffic from the ISP in autonomous system 45000.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]

6. **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. Repeat Step 7 to apply another route map to the neighbor specified in Step 7.
9. **exit**
10. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
11. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
12. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
13. Repeat Step 10 to apply another route map to the neighbor specified in Step 10.
14. **exit**
15. **exit**
16. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
17. Repeat Step 14 to establish another static route.
18. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
19. **set metric** *value*
20. **exit**
21. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
22. **set metric** *value*
23. **exit**
24. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
25. **end**
26. **show ip route** [*ip-address*] [*mask*] [**longer-prefixes**]
27. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:**	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

	Command or Action	Purpose
	`Router(config-router)# neighbor 192.168.2.1 remote-as 45000`	• In this example, the BGP peer at Router D is added to the BGP routing table.
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] **Example:** `Router(config-router-af)# network 10.2.2.0 mask 255.255.255.0`	Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.

Step 7	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
	Example:
	`Router(config-router-af)# neighbor 192.168.2.1 route-map BLOCK in`
	Example:
	`Router(config-router-af)# neighbor 192.168.2.1 route-map SETMETRIC1 out`
		Applies a route map to incoming or outgoing routes.
		• In the first example, the route map named BLOCK is applied to inbound routes at Router E.
		• In the second example, the route map named SETMETRIC1 is applied to outbound routes to Router D.
		Note Two examples are shown here because the task example requires both these statements to be configured.

Step 8	Repeat Step 7 to apply another route map to the neighbor specified in Step 7.	--
Step 9	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.

	Command or Action	Purpose
Step 10	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.3.1 remote-as 45000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • In this example, the BGP peer at Router D is added to the BGP routing table.
Step 11	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

Step 12	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
	Example:
	`Router(config-router-af)# neighbor 192.168.3.1 route-map BLOCK in`
	Example:
	`Router(config-router-af)# neighbor 192.168.3.1 route-map SETMETRIC2 out`
		Applies a route map to incoming or outgoing routes.
		• In the first example, the route map named BLOCK is applied to inbound routes at Router E.
		• In the second example, the route map named SETMETRIC2 is applied to outbound routes to Router D.
		Note Two examples are shown here because the task example requires both these statements to be configured.

Step 13	Repeat Step 10 to apply another route map to the neighbor specified in Step 10.	--
Step 14	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 15	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters global configuration mode.

	Command or Action	Purpose

Step 16	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50`
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1 50`
	Example:
	`and`
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 192.168.3.1 40`
		Establishes a static route.
		• In the first example, a static route to BGP peer 192.168.2.1 is established and given an administrative distance of 50.
		• In the second example, a static route to BGP peer 192.168.3.1 is established and given an administrative distance of 40. The lower administrative distance makes this route via Router B the preferred route.
		Note Two examples are shown here because the task example requires both these statements to be configured.

Step 17	Repeat Step 14 to establish another static route.	--
Step 18	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map SETMETRIC1 permit 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named SETMETRIC1 is created.
Step 19	**set metric** *value* **Example:** `Router(config-route-map)# set metric 100`	Sets the MED metric value.
Step 20	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 21	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map SETMETRIC2 permit 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named SETMETRIC2 is created.
Step 22	**set metric** *value* **Example:** `Router(config-route-map)# set metric 50`	Sets the MED metric value.

	Command or Action	Purpose
Step 23	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 24	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map BLOCK deny 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named BLOCK is created to block all incoming routes from autonomous system 45000.
Step 25	**end** **Example:** `Router(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.
Step 26	**show ip route** [*ip-address*] [*mask*] [**longer-prefixes**] **Example:** `Router# show ip route`	(Optional) Displays route information from the routing tables. • Use this command at Router E in the figure above after Router B and Router D have received update information containing the MED metric from Router E. • Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 27	**show ip bgp** [*network*] [*network-mask*] **Example:** `Router# show ip bgp 172.17.1.0 255.255.255.0`	(Optional) Displays the entries in the BGP routing table. • Use this command at Router E in the figure above after Router B and Router D have received update information containing the MED metric from Router E. • Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following example shows output from the **show ip route** command entered at Router E after this task has been configured and Router B and Router D have received update information containing the MED metric. Note that the gateway of last resort is set as 192.168.3.1, which is the route to Router B.

```
Router# show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 192.168.3.1 to network 0.0.0.0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.2.2.0 is directly connected, Ethernet0/0
C    192.168.2.0/24 is directly connected, Serial3/0
C    192.168.3.0/24 is directly connected, Serial2/0
S*   0.0.0.0/0 [40/0] via 192.168.3.1
```

The following example shows output from the **show ip bgp** command entered at Router E after this task has been configured and Router B and Router D have received routing updates. The route map BLOCK has denied all routes coming in from autonomous system 45000 so the only network shown is the local network.

```
Router# show ip bgp

BGP table version is 2, local router ID is 10.2.2.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      0.0.0.0                  0         32768 i
```

The following example shows output from the **show ip bgp** command entered at Router B after this task has been configured at Router E and Router B has received routing updates. Note the metric of 50 for network 10.2.2.0.

```
Router# show ip bgp

BGP table version is 7, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 40000 i
*> 10.2.2.0/24      192.168.3.2             50             0 50000 i
*> 172.16.1.0/24    0.0.0.0                  0         32768 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

The following example shows output from the **show ip bgp** command entered at Router D after this task has been configured at Router E and Router D has received routing updates. Note the metric of 100 for network 10.2.2.0.

```
Router# show ip bgp

BGP table version is 3, local router ID is 192.168.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.2.2             100             0 50000 i
*> 172.16.1.0/24    0.0.0.0                  0         32768 i
```

Configuring Multihoming to Receive the Full Internet Routing Table

Perform this task to configure your network to build neighbor relationships with other routers in other autonomous systems while filtering outbound routes. In this task the full Internet routing table will be received from the service providers in the neighboring autonomous systems but only locally originated routes will be advertised to the service providers. This task is configured at Router B in the figure above and uses an access list to permit only locally originated routes and a route map to ensure that only the locally originated routes are advertised outbound to other autonomous systems.

Note Be aware that receiving the full Internet routing table from two ISPs may use all the memory in smaller routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **network** *network-number* [**mask** *network-mask*]
7. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
11. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
12. **exit**
13. **exit**
14. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
15. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
16. **match as-path** *path-list-number*
17. **end**
18. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**network** *network-number* [**mask** *network-mask*] **Example:** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Router(config-router-af)# neighbor 192.168.1.2 route-map localonly out`	Applies a route map to incoming or outgoing routes. • In this example, the route map named localonly is applied to outbound routes to Router A.
Step 8	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.

	Command or Action	Purpose
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.3.2 remote-as 50000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 10	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Router(config-router-af)# neighbor 192.168.3.2 route-map localonly out`	Applies a route map to incoming or outgoing routes. • In this example, the route map named localonly is applied to outbound routes to Router E.
Step 12	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 13	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 14	**ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression* **Example:** `Router(config)# ip as-path access-list 10 permit ^$`	Defines a BGP-related access list. • In this example, the access list number 10 is defined to permit only locally originated BGP routes.
Step 15	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*]	Configures a route map and enters route map configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# route-map localonly permit 10`	• In this example, a route map named localonly is created.
Step 16	**match as-path** *path-list-number* **Example:** `Router(config-route-map)# match as-path 10`	Matches a BGP autonomous system path access list. • In this example, the BGP autonomous system path access list created in Step 12 is used for the match clause.
Step 17	**end** **Example:** `Router(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.
Step 18	**show ip bgp** [*network*] [*network-mask*] **Example:** `Router# show ip bgp`	Displays the entries in the BGP routing table. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following example shows the BGP routing table for Router B in the figure above after this task has been configured. Note that the routing table contains the information about the networks in the autonomous systems 40000 and 50000.

```
BGP table version is 5, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.2              0             0 40000 i
*> 10.2.2.0/24      192.168.3.2              0             0 50000 i
*> 172.17.1.0/24    0.0.0.0                  0         32768 i
```

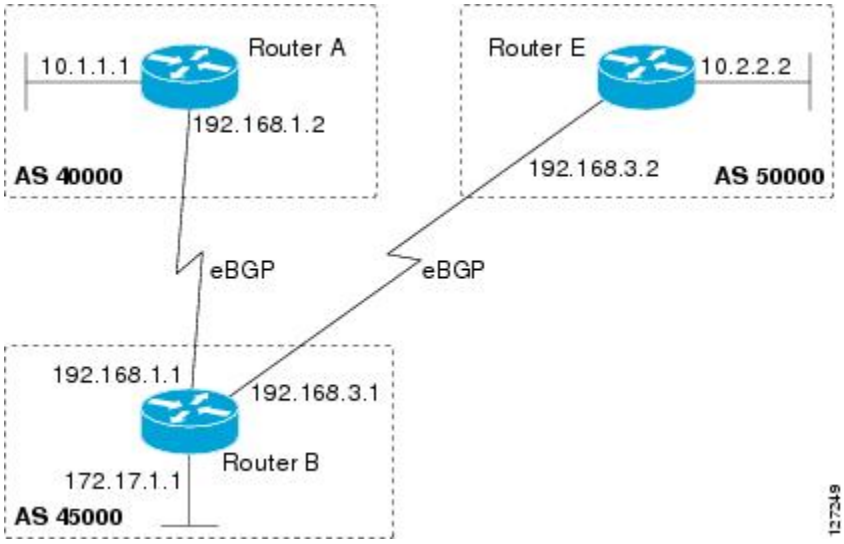
Configuring BGP Policies

The tasks in this section help you configure BGP policies that filter the traffic in your BGP network. The following optional tasks demonstrate some of the various methods by which traffic can be filtered in your BGP network:

Filtering BGP Prefixes with Prefix Lists

Perform this task to use prefix lists to filter BGP route information. The task is configured at Router B in the figure below where both Router A and Router E are set up as BGP peers. A prefix list is configured to permit only routes from the network 10.2.2.0/24 to be outbound. In effect, this will restrict the information that is received from Router E to be forwarded to Router A. Optional steps are included to display the prefix list information and to reset the hit count.

Figure 59: BGP Topology for Configuring BGP Policies Tasks



Note The **neighbor prefix-list** and the **neighbor distribute-list** commands are mutually exclusive for a BGP peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. Repeat Step 5 for all BGP peers.
6. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*]
8. **aggregate-address** *address mask* [**as-set**]
9. **neighbor** *ip-address* **prefix-list** *list-name* {**in** | **out**}
10. **exit**
11. **exit**
12. **ip prefix-list** *list-name* [**seq** *seq-number*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] [**eq** *eq-value*]
13. **end**
14. **show ip prefix-list** [**detail** | **summary**] [*prefix-list-name* [**seq** *seq-number* | *network/length* [**longer** | **first-match**]]]
15. **clear ip prefix-list** {***** | *ip-address* | *peer-group-name*} **out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address of the neighbor in the specified autonomous system to the BGP neighbor table of the local router.
Step 5	Repeat Step 5 for all BGP peers.	--
Step 6	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 7	**network** *network-number* [**mask** *network-mask*] **Example:** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 8	**aggregate-address** *address mask* [**as-set**] **Example:**	Creates an aggregate entry in a BGP routing table. • A specified route must exist in the BGP table.

	Command or Action	Purpose
	`Router(config-router-af)# aggregate-address 172.0.0.0 255.0.0.0`	• Use the **aggregate-address** command with no keywords to create an aggregate entry if any more-specific BGP routes are available that fall in the specified range. **Note** Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 9	**neighbor** *ip-address* **prefix-list** *list-name* {**in** | **out**} **Example:** `Router(config-router-af)# neighbor 192.168.1.2 prefix-list super172 out`	Distributes BGP neighbor information as specified in a prefix list. • In this example, a prefix list called super172 is set for outgoing routes to Router A.
Step 10	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 11	**exit** **Example:** `Router(config-router) exit`	Exits router configuration mode and enters global configuration mode.
Step 12	**ip prefix-list** *list-name* [**seq** *seq-number*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] [**eq** *eq-value*] **Example:** `Router(config)# ip prefix-list super172 permit 172.0.0.0/8`	Defines a BGP-related prefix list and enters access list configuration mode. • In this example, the prefix list called super172 is defined to permit only route 172.0.0.0/8 to be forwarded. • All other routes will be denied because there is an implicit deny at the end of all prefix lists.
Step 13	**end** **Example:** `Router(config-access-list)# end`	Exits access list configuration mode and enters privileged EXEC mode.
Step 14	**show ip prefix-list** [**detail** | **summary**] [*prefix-list-name* [**seq** *seq-number* | *network/length* [**longer** | **first-match**]]] **Example:** `Router# show ip prefix-list detail super172`	Displays information about prefix lists. • In this example, details of the prefix list named super172 will be displayed, including the hit count. Hit count is the number of times the entry has matched a route.
Step 15	**clear ip prefix-list** {***** | *ip-address* | *peer-group-name*} **out**	Resets the hit count of the prefix list entries.

	Command or Action	Purpose
	Example: `Router# clear ip prefix-list super172 out`	• In this example, the hit count for the prefix list called super172 will be reset.

Examples

The following output from the **show ip prefix-list** command shows details of the prefix list named super172, including the hit count. The **clear ip prefix-list** command is entered to reset the hit count and the **show ip prefix-list** command is entered again to show the hit count reset to 0.

```
Router# show ip prefix-list detail super172

ip prefix-list super172:
   count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
   seq 5 permit 172.0.0.0/8 (hit count: 1, refcount: 1)

Router# clear ip prefix-list super172

Router# show ip prefix-list detail super172

ip prefix-list super172:
   count: 1, range entries: 0, sequences: 5 - 5, refcount: 4
   seq 5 permit 172.0.0.0/8 (hit count: 0, refcount: 1)
```

Filtering BGP Prefixes with AS-Path Filters

Perform this task to filter BGP prefixes using AS-path filters with an access list based on the value of the AS-path attribute to filter route information. An AS-path access list is configured at Router B in the figure above. The first line of the access list denies all matches to AS-path 50000, and the second line allows all other paths. The router uses the **neighbor filter-list** command to specify the AS-path access list as an outbound filter. After the filter is enabled, traffic can be received from both Router A and Router C, but updates originating from autonomous system 50000 (Router C) are not forwarded by Router B to Router A. If any updates from Router C originated from another autonomous system, they would be forwarded because they would contain both autonomous system 50000 and another autonomous system number, and that would not match the AS-path access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
4. Repeat Step 3 for all entries required in the AS-path access list.
5. **router bgp** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
7. Repeat Step 6 for all BGP peers.
8. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
9. **neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out**}
10. **end**
11. **show ip bgp regexp** *as-regular-expression*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
	Example:
	`Device(config)# ip as-path access-list 100 deny ^50000$`
	Example:
	`Device(config)# ip as-path access-list 100 permit .*`
		Defines a BGP-related access list and enters access list configuration mode.
		• In the first example, access list number 100 is defined to deny any AS-path that starts and ends with 50000.
		• In the second example, all routes that do not match the criteria in the first example of the AS-path access list will be permitted. The period and asterisk symbols imply that all characters in the AS-path will match, so Router B will forward those updates to Router A.
		Note Two examples are shown here because the task example requires both these statements to be configured.

Step 4	Repeat Step 3 for all entries required in the AS-path access list.	—
Step 5	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system BGP neighbor table of the local router.
Step 7	Repeat Step 6 for all BGP peers.	—
Step 8	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4

	Command or Action	Purpose
		unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 192.168.1.2 filter-list 100 out`	Distributes BGP neighbor information as specified in a prefix list. • In this example, an access list number 100 is set for outgoing routes to Router A.
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 11	**show ip bgp regexp** *as-regular-expression* **Example:** `Device# show ip bgp regexp ^50000$`	Displays routes that match the regular expression. • To verify the regular expression, you can use this command. • In this example, all paths that match the expression "starts and ends with 50000" will be displayed.

Examples

The following output from the **show ip bgp regexp** command shows the autonomous system paths that match the regular expression—start and end with AS-path 50000:

```
Device# show ip bgp regexp ^50000$

BGP table version is 9, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.3.2              0           150 50000 i
```

Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers

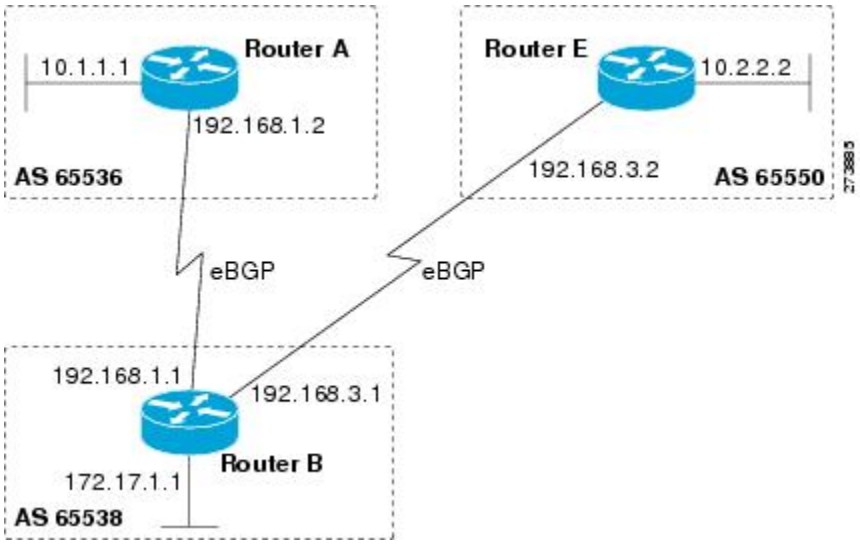
In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXI1, and later releases, BGP support for 4-octet (4-byte) autonomous system numbers was introduced. The 4-byte autonomous system numbers in this task are formatted in the default asplain (decimal value) format, for example, Router B is in autonomous

system number 65538 in the figure below. For more details about the introduction of 4-byte autonomous system numbers, see the "BGP Autonomous System Number Formats" section.

Perform this task to filter BGP prefixes with AS-path filters using 4-byte autonomous system numbers with an access list based on the value of the AS-path attribute to filter route information. An AS-path access list is configured at Router B in the figure below. The first line of the access list denies all matches to the AS-path 65550 and the second line allows all other paths. The router uses the **neighbor filter-list** command to specify the AS-path access list as an outbound filter. After the filtering is enabled, traffic can be received from both Router A and Router E but updates originating from autonomous system 65550 (Router E) are not forwarded by Router B to Router A. If any updates from Router E originated from another autonomous system, they would be forwarded because they would contain both autonomous system 65550 plus another autonomous system number, and that would not match the AS-path access list.

Note In Cisco IOS Releases 12.0(22)S, 12.2(15)T, 12.2(18)S, and later releases, the maximum number of autonomous system access lists that can be configured with the **ip as-path access-list** command is increased from 199 to 500.

Figure 60: BGP Topology for Filtering BGP Prefixes with AS-path Filters Using 4-Byte Autonomous System Numbers



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. Repeat Step 4 for all BGP peers.
6. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
7. **network** *network-number* [**mask** *network-mask*]
8. **neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number*{**in** | **out**}
9. **exit**
10. **exit**

11. **ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
12. Repeat Step 11 for all entries required in the AS-path access list.
13. **end**
14. **show ip bgp regexp** *as-regular-expression*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65538`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.2 remote-as 65536`	Adds the IP address or peer group name of the neighbor in the specified autonomous system BGP neighbor table of the local router. • In this example, the IP address for the neighbor at Router A is added.
Step 5	Repeat Step 4 for all BGP peers.	--
Step 6	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

	Command or Action	Purpose
Step 7	**network** *network-number* [**mask** *network-mask*] **Example:** `Router(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates. **Note** Only partial syntax is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number*{**in** | **out**} **Example:** `Router(config-router-af)# neighbor 192.168.1.2 filter-list 99 out`	Distributes BGP neighbor information as specified in a prefix list. • In this example, an access list number 99 is set for outgoing routes to Router A.
Step 9	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 10	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.

Step 11	**ip as-path access-list** *access-list-number* {**deny** | **permit**} *as-regular-expression*
	Example:
	`Router(config)# ip as-path access-list 99 deny ^65550$`
	Example:
	`and`
	Example:
	`Router(config)# ip as-path access-list 99 permit .*`
		Defines a BGP-related access list and enters access list configuration mode.
		• In the first example, access list number 99 is defined to deny any AS-path that starts and ends with 65550.
		• In the second example, all routes that do not match the criteria in the first example of the AS-path access list will be permitted. The period and asterisk symbols imply that all characters in the AS-path will match, so Router B will forward those updates to Router A.
		Note Two examples are shown here because the task example requires both these statements to be configured.

Step 12	Repeat Step 11 for all entries required in the AS-path access list.	--
Step 13	**end** **Example:**	Exits access list configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config-access-list)# end`	
Step 14	**show ip bgp regexp** *as-regular-expression* **Example:** `Router# show ip bgp regexp ^65550$`	Displays routes that match the regular expression. • To verify the regular expression, you can use this command. • In this example, all paths that match the expression "starts and ends with 65550" will be displayed.

Examples

The following output from the **show ip bgp regexp** command shows the autonomous system paths that match the regular expression--start and end with AS-path 65550:

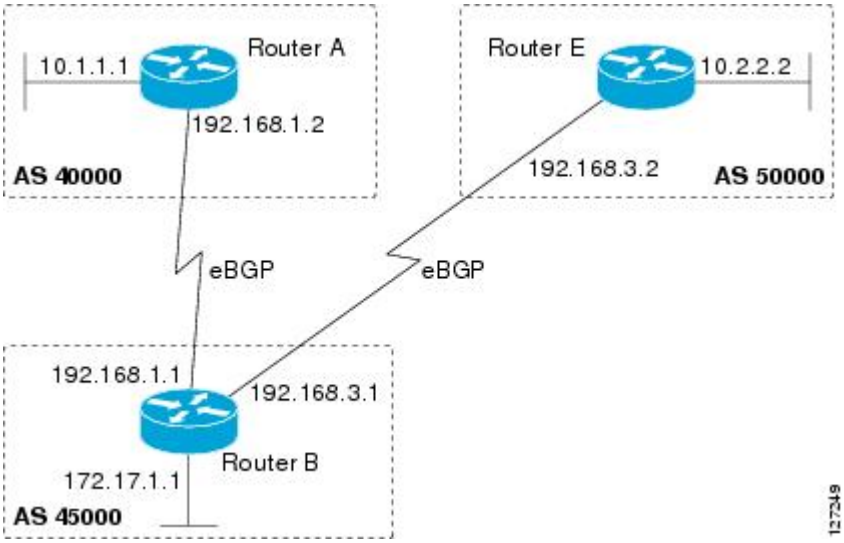
```
RouterB# show ip bgp regexp ^65550$
BGP table version is 4, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.2.2.0/24      192.168.3.2              0             0 65550 i
```

Filtering Traffic Using Community Lists

Perform this task to filter traffic by creating a BGP community list, referencing the community list within a route map, and then applying the route map to a neighbor.

In this task, Router B in the figure below is configured with route maps and a community list to control incoming routes.

Figure 61: Topology for Which a Community List Is Configured



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
10. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
11. **set weight** *weight*
12. **exit**
13. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
14. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
15. **set community** *community-number*
16. **exit**
17. **ip community-list** {*standard-list-number* | **standard** *list-name* {**deny** | **permit**} [*community-number*] [*AA:NN*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**]} | {*expanded-list-number* | **expanded** *list-name* {**deny** | **permit**} *regular-expression*}
18. Repeat Step 17 to create all the required community lists.
19. **exit**
20. **show ip community-list** [*standard-list-number* | *expanded-list-number* | *community-list-name*] [**exact-match**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:**	Adds the IP address or peer group name of the neighbor to the specified autonomous system BGP neighbor table of the local router.

	Command or Action	Purpose
	`Device(config-router)# neighbor 192.168.3.2 remote-as 50000`	
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 192.168.3.2 route-map 2000 in`	Applies a route map to inbound or outbound routes. • In this example, the route map called 2000 is applied to inbound routes from the BGP peer at 192.168.3.2.
Step 7	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 8	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 9	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map 2000 permit 10`	Creates a route map and enters route map configuration mode. • In this example, the route map called 2000 is defined.
Step 10	**match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]} **Example:** `Device(config-route-map)# match community 1`	Matches on the communities in a BGP community list. • In this example, the route's community attribute is matched to communities in community list 1.

	Command or Action	Purpose
Step 11	**set weight** *weight* **Example:** `Device(config-route-map)# set weight 30`	Sets the weight of BGP routes that match the community list. • In this example, any route that matches community list 1 will have its weight set to 30.
Step 12	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 13	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map 3000 permit 10`	Creates a route map and enters route map configuration mode. • In this example, the route map called 3000 is defined.
Step 14	**match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]} **Example:** `Device(config-route-map)# match community 2`	Matches on the communities in a BGP community list. • In this example, the route's COMMUNITIES attribute is matched to communities in community list 2.
Step 15	**set community** *community-number* **Example:** `Device(config-route-map)# set community 99`	Sets the BGP communities attribute. • In this example, any route that matches community list 2 will have the COMMUNITIES attribute set to 99.
Step 16	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.

Step 17	**ip community-list** {*standard-list-number* | **standard** *list-name* {**deny** | **permit**} [*community-number*] [*AA*:*NN*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**]} | {*expanded-list-number* | **expanded** *list-name* {**deny** | **permit**} *regular-expression*}
	Example:
	`Device(config)# ip community-list 1 permit 100`
	Example:
	`Device(config)# ip community-list 2 permit internet`
		Creates a community list for BGP and controls access to it.
		• In the first example, community list 1 permits routes with a COMMUNITIES attribute of 100. Router E routes all have a COMMUNITIES attribute of 100, so their weight will be set to 30.
		• In the second example, community list 2 effectively permits all routes by specifying the **internet** community. Any routes that did not match community list 1 are checked against community list 2. All routes are permitted, but no changes are made to the route attributes.

	Command or Action	Purpose
		Note Two examples are shown here because the task example requires both of these statements to be configured.
Step 18	Repeat Step 17 to create all the required community lists.	—
Step 19	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 20	**show ip community-list** [*standard-list-number* | *expanded-list-number* | *community-list-name*] [**exact-match**] **Example:** `Device# show ip community-list 1`	Displays configured BGP community list entries.

Examples

The following sample output verifies that community list 1 has been created and it permits routes that have a community attribute of 100:

```
Device# show ip community-list 1

Community standard list 1
     permit 100
```

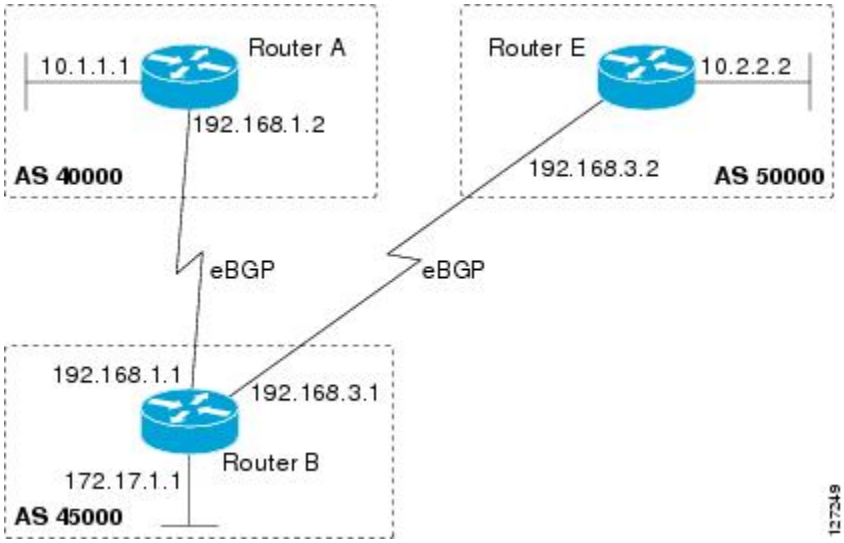
The following sample output verifies that community list 2 has been created and it effectively permits all routes by specifying the **internet** community:

```
Device# show ip community-list 2

Community standard list 2
     permit internet
```

Filtering Traffic Using Extended Community Lists

Perform this task to filter traffic by creating an extended BGP community list to control outbound routes.

Figure 62: Topology for Which a Community List Is Configured



In this task, Router B in the figure above is configured with an extended named community list to specify that the BGP peer at 192.168.1.2 is not sent advertisements about any path through or from autonomous system 50000. The IP extended community-list configuration mode is used and the ability to resequence entries is shown.

Note A sequence number is applied to all extended community list entries by default, regardless of the configuration mode. Explicit sequencing and resequencing of extended community list entries can be configured only in IP extended community-list configuration mode, not in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** {*expanded-list-number* | **expanded** *list-name* | *standard-list-number* | **standard** *list-name*}
4. [*sequence-number*] {**deny** [*regular-expression*] | **exit** | **permit** [*regular-expression*]}
5. Repeat Step 4 for all the required permit or deny entries in the extended community list.
6. **resequence** [*starting-sequence*] [*sequence-increment*]
7. **exit**
8. **router bgp** *autonomous-system-number*
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
10. Repeat the prior step for all of the required BGP peers.
11. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
12. **network** *network-number* [**mask** *network-mask*]
13. **end**
14. **show ip extcommunity-list** [*list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** {*expanded-list-number* | **expanded** *list-name* | *standard-list-number* | **standard** *list-name*} **Example:** `Device(config)# ip extcommunity-list expanded DENY50000`	Enters IP extended community-list configuration mode to create or configure an extended community list. • In this example, the expanded community list DENY50000 is created.

Step 4	[*sequence-number*] {**deny** [*regular-expression*] | **exit** | **permit** [*regular-expression*]}
	Example:
	`Device(config-extcomm-list)# 10 deny _50000_`
	Example:
	`Device(config-extcomm-list)# 20 deny ^50000 .*`
		Configures an expanded community list entry.
		• In the first example, an expanded community list entry with the sequence number 10 is configured to deny advertisements about paths from autonomous system 50000.
		• In the second example, an expanded community list entry with the sequence number 20 is configured to deny advertisements about paths through autonomous system 50000.
		Note Two examples are shown here because the task example requires both these statements to be configured.
		Note Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Step 5	Repeat Step 4 for all the required permit or deny entries in the extended community list.	—
Step 6	**resequence** [*starting-sequence*] [*sequence-increment*] **Example:** `Device(config-extcomm-list)# resequence 50 100`	Resequences expanded community list entries. • In this example, the sequence number of the first expanded community list entry is set to 50 and subsequent entries are set to increment by 100. The second expanded community list entry is therefore set to 150.

	Command or Action	Purpose
		Note Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 7	**exit** **Example:** `Device(config-extcomm-list)# exit`	Exits expanded community-list configuration mode and enters global configuration mode.
Step 8	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.3.2 remote-as 50000`	Adds the IP address or peer group name of the neighbor to the specified autonomous system BGP neighbor table of the local router.
Step 10	Repeat the prior step for all of the required BGP peers.	—
Step 11	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified in the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. **Note** The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 12	**network** *network-number* [**mask** *network-mask*] **Example:** `Device(config-router-af)# network 172.17.1.0 mask 255.255.255.0`	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. • For exterior protocols, the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.

	Command or Action	Purpose
		Note Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 13	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 14	**show ip extcommunity-list** [*list-name*] **Example:** `Device# show ip extcommunity-list DENY50000`	Displays configured BGP expanded community list entries.

Examples

The following sample output verifies that the BGP expanded community list DENY50000 has been created, with the output showing that the entries to deny advertisements about autonomous system 50000 have been resequenced from 10 and 20 to 50 and 150:

```
Device# show ip extcommunity-list DENY50000

Expanded extended community-list DENY50000
     50 deny _50000_
     150 deny ^50000 .*
```

Filtering Traffic Using a BGP Route Map Policy List

Perform this task to create a BGP policy list and then reference it within a route map.

A policy list is like a route map that contains only match clauses. With policy lists there are no changes to match clause semantics and route map functions. The match clauses are configured in policy lists with permit and deny statements and the route map evaluates and processes each match clause to permit or deny routes based on the configuration. AND and OR semantics in the route map function the same way for policy lists as they do for match clauses.

Policy lists simplify the configuration of BGP routing policy in medium-size and large networks. The network operator can reference preconfigured policy lists with groups of match clauses in route maps and easily apply general changes to BGP routing policy. The network operator no longer needs to manually reconfigure each recurring group of match clauses that occur in multiple route map entries.

Perform this task to create a BGP policy list to filter traffic that matches the autonomous system path and MED of a router and then create a route map to reference the policy list.

Before you begin

BGP routing must be configured in your network and BGP neighbors must be established.

Note

- BGP route map policy lists do not support the configuration of IPv6 match clauses in policy lists.
- Policy lists are not supported in versions of Cisco IOS software prior to Cisco IOS Releases 12.0(22)S and 12.2(15)T. Reloading a router that is running an older version of Cisco IOS software may cause some routing policy configurations to be lost.
- Policy lists support only match clauses and do not support set clauses. However, policy lists can coexist, within the same route map entry, with match and set clauses that are configured separately from the policy lists.
- Policy lists are supported only by BGP. They are not supported by other IP routing protocols. This limitation does not interfere with normal operations of a route map, including redistribution, because policy list functions operate transparently within BGP and are not visible to other IP routing protocols.
- Policy lists support only match clauses and do not support set clauses. However, policy lists can coexist, within the same route map entry, with match and set clauses that are configured separately from the policy lists. The first route map example configures AND semantics, and the second route map configuration example configures semantics. Both examples in this section show sample route map configurations that reference policy lists and separate match and set clauses in the same configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip policy-list** *policy-list-name* {**permit** | **deny**}
4. **match as-path** *as-number*
5. **match metric** *metric*
6. **exit**
7. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
8. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
9. **match policy-list** *policy-list-name*
10. **set community** *community-number* [**additive**] [*well-known-community*] | **none**}
11. **set local-preference** *preference-value*
12. **end**
13. **show ip policy-list** [*policy-list-name*]
14. **show route-map** [*route-map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip policy-list** *policy-list-name* {**permit** | **deny**} **Example:** `Router(config)# ip policy-list POLICY-LIST-NAME-1 permit`	Enters policy list configuration mode and creates a BGP policy list that will permit routes that are allowed by the match clauses that follow.
Step 4	**match as-path** *as-number* **Example:** `Router(config-policy-list)# match as-path 500`	Creates a match clause to permit routes from the specified autonomous system path.
Step 5	**match metric** *metric* **Example:** `Router(config-policy-list)# match metric 10`	Creates a match clause to permit routes with the specified metric.
Step 6	**exit** **Example:** `Router(config-policy-list)# exit`	Exits policy list configuration mode and enters global configuration mode.
Step 7	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map MAP-NAME-1 permit 10`	Creates a route map and enters route map configuration mode.
Step 8	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Router(config-route-map)# match ip address 1`	Creates a match clause to permit routes that match the specified *access-list-number*or *access-list-name*argument.
Step 9	**match policy-list** *policy-list-name* **Example:** `Router(config-route-map)# match policy-list POLICY-LIST-NAME-1`	Creates a clause that will match the specified policy list. • All match clauses within the policy list will be evaluated and processed. Multiple policy lists can referenced with this command. • This command also supports AND or OR semantics like a standard match clause.
Step 10	**set community** *community-number* [**additive**] [*well-known-community*] | **none**} **Example:** `Router(config-route-map)# set community 10:1`	Creates a clause to set or remove the specified community.

	Command or Action	Purpose
Step 11	**set local-preference** *preference-value* **Example:** `Router(config-route-map)# set local-preference 140`	Creates a clause to set the specified local preference value.
Step 12	**end** **Example:** `Router(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.
Step 13	**show ip policy-list** [*policy-list-name*] **Example:** `Router# show ip policy-list POLICY-LIST-NAME-1`	Display information about configured policy lists and policy list entries.
Step 14	**show route-map** [*route-map-name*] **Example:** `Router# show route-map`	Displays locally configured route maps and route map entries.

Examples

The following sample output verifies that a policy list has been created, with the output displaying the policy list name and configured match clauses:

```
Router# show ip policy-list
 POLICY-LIST-NAME-1

policy-list POLICY-LIST-NAME-1 permit
  Match clauses:
    metric 20
    as-path (as-path filter): 1
```

Note A policy list name can be specified when the **show ip policy-list** command is entered. This option can be useful for filtering the output of this command and verifying a single policy list.

The following sample output from the **show route-map** command verifies that a route map has been created and a policy list is referenced. The output of this command displays the route map name and policy lists that are referenced by the configured route maps.

```
Router# show route-map

route-map ROUTE-MAP-NAME-1, deny, sequence 10
  Match clauses:
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map ROUTE-MAP-NAME-1, permit, sequence 10
```

```
Match clauses:
  IP Policy lists:
      POLICY-LIST-NAME-1
Set clauses:
Policy routing matches: 0 packets, 0 bytes
```

Filtering Traffic Using Continue Clauses in a BGP Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
10. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
11. **set community** { { [*community-number*] [*well-known-community*] [**additive**]} | **none**}
12. **continue** [*sequence-number*]
13. **end**
14. **show route-map** [*map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:**	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.

	Command or Action	Purpose
	`Device(config-router)# neighbor 10.0.0.1 remote-as 50000`	
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the device is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in`	Applies the inbound route map to routes received from the specified neighbor, or applies an outbound route map to routes advertised to the specified neighbor.
Step 7	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 8	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 9	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME permit 10`	Enters route-map configuration mode to create or configure a route map.
Step 10	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Device(config-route-map)# match ip address 1`	Configures a **match** command that specifies the conditions under which policy routing and route filtering occur. • Multiple **match** commands can be configured. If a **match** command is configured, a match must occur in order for the continue statement to be executed. If

	Command or Action	Purpose
		a **match** command is not configured, set and continue clauses will be executed. **Note** The **match** and **set** commands used in this task are examples that are used to help describe the operation of the **continue** command. For a list of specific **match** and **set** commands, see the **continue** command in the *Cisco IOS IP Routing: BGP Command Reference*.
Step 11	**set community** { { [*community-number*] [*well-known-community*] [**additive**]} | **none**} **Example:** `Device(config-route-map)# set community 10:1`	Configures a **set** command that specifies the routing action to perform if the criteria enforced by the **match** commands are met. • Multiple **set** commands can be configured. • In this example, a clause is created to set the specified community number in aa:nn format.
Step 12	**continue** [*sequence-number*] **Example:** `Device(config-route-map)# continue`	Configures a route map to continue to evaluate and execute match statements after a successful match occurs. • If a sequence number is configured, the continue clause will go to the route map with the specified sequence number. • If no sequence number is specified, the continue clause will go to the route map with the next sequence number. This behavior is called an "implied continue."
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	**show route-map** [*map-name*] **Example:** `Device# show route-map`	(Optional) Displays locally configured route maps. The name of the route map can be specified in the syntax of this command to filter the output.

Examples

The following sample output shows how to verify the configuration of continue clauses using the **show route-map** command. The output displays configured route maps including the match, set, and continue clauses.

```
Device# show route-map
```

```
route-map MARKETING, permit, sequence 10
  Match clauses:
    ip address (access-lists): 1
    metric 10
  Continue: sequence 40
  Set clauses:
    as-path prepend 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 20
  Match clauses:
    ip address (access-lists): 2
    metric 20
  Set clauses:
    as-path prepend 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 30
  Match clauses:
  Continue: to next entry 40
  Set clauses:
    as-path prepend 10 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 40
  Match clauses:
    community (community-list filter): 10:1
  Set clauses:
    local-preference 104
  Policy routing matches: 0 packets, 0 bytes
route-map MKTG-POLICY-MAP, permit, sequence 10
  Match clauses:
  Set clauses:
    community 655370
  Policy routing matches: 0 packets, 0 bytes
```

Configuring EBGP Route Propagation without Policies

To prevent EBGP route propagation by a router when at least one inbound and one outbound policy are not configured, use the **bgp safe-ebgp-policy** configuration option.

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **router bgp** *autonomous-system-number*

Example:

```
Router(config)# router bgp 1
```

Configures a BGP routing process, and enters router configuration mode for the specified routing process.

Step 4 **bgp safe-ebgp-policy**

Example:

```
Router(config-router)# bgp safe-ebgp-policy
```

Configures the BGP routing to process to not accept route announcements from or advertise route announcements to an EBGP neighbor unless at least one inbound and one outbound policy are configured for the neighbor.

Step 5 **end**

Example:

```
Router(config-router)# end
```

Exits router configuration mode and enters privileged EXEC mode.

Verifying EBGP Route Propagation without Policies

If **bgp safe-ebgp-policy** is configured, and at least one inbound and one outbound policy are not configured for an EBGP neighbor, an EBGP router does not propagate routes to or from the neighbor. The router verifies whether inbound and outbound policies are configured for each address family. If the policies are configured for an address family, say IPv4, and not for another address family, say IPv6, the router propagates routes for the address family for which policies are configured.

Verifying Route Propagation for a Specific Address Family

You can verify the EBGP route propagation behavior for an address family using the command **show ip bgp** *address-family* **summary**.

The following sample output is from a router on which **bgp safe-ebgp-policy** is configured, and inbound and outbound policies are not configured for the neighbor for the address family IPv4 Unicast. `0!` indicates that routes have not been accepted from or advertised to the neighbor .

```
R1#show ip bgp ipv4 unicast summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.0.0.2       4            2       4       6        1    0    0 00:00:11        0!
```

Verifying Route Propagation for a Specific Address Family and Neighbor

You can verify the EBGP route propagation behavior for an address family and for a particular neighbor using the command **show ip bgp** *address-family* neighbor *ip-address*.

The following sample output is from a router on which **bgp safe-ebgp-policy** is configured, and inbound and outbound policies are not configured for the neighbor for the address family IPv4 Unicast. The highlighted statement `Suppressing inbound/outbound propagation because policies are missing` indicates that EBGP routes are not being propagated.

```
R1#show ip bgp ipv4 unicast neighbor 192.0.0.2
BGP neighbor is 192.0.0.2,  remote AS 2, external link
  BGP version 4, remote router ID 1.1.1.2
  BGP state = Established, up for 00:02:09
  Last read 00:00:33, last write 00:00:31, hold time is 180, keepalive interval is 60 seconds

  Neighbor sessions:
```

```
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          1
    Keepalives:             3          3
    Route Refresh:          1          0
    Total:                  8          7
  Do log neighbor state changes (via global configuration)
  Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
  Session: 192.0.0.2
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 1, Advertise bit 0
  1 update-group member
  Suppressing inbound/outbound propagation because policies are missing
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
                                 Sent       Rcvd
  Prefix activity:               ----       ----
    Prefixes Current:               0          0
    Prefixes Total:                 0          0
    Implicit Withdraw:              0          0
    Explicit Withdraw:              0          0
    Used as bestpath:             n/a          0
    Used as multipath:            n/a          0
    Used as secondary:            n/a          0

                                   Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    Total:                                0          0
  Number of NLRIs in the update sent: max 0, min 0
  Last detected as dynamic slow peer: never
  Dynamic slow peer recovered: never
  Refresh Epoch: 2
  Last Sent Refresh Start-of-rib: 00:02:01
  Last Sent Refresh End-of-rib: 00:02:01
  Refresh-Out took 0 seconds
                                     Sent       Rcvd
        Refresh activity:            ----       ----
          Refresh Start-of-RIB          1          1
          Refresh End-of-RIB            1          1

  Address tracking is enabled, the RIB does have a route to 192.0.0.2
  Route to peer address reachability Up: 1; Down: 0
    Last notification 00:02:09
  Connections established 1; dropped 0
  Last reset never
  External BGP neighbor configured for connected checks (single-hop
no-disable-connected-check)
  Interface associated: Ethernet0/0 (peering address in same link)
```

```
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
  SSO is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 1
Local host: 192.0.0.1, Local port: 179
Foreign host: 192.0.0.2, Foreign port: 27432
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x45C8D3A):
Timer          Starts    Wakeups            Next
Retrans             7          0             0x0
TimeWait            0          0             0x0
AckHold             5          2             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
ProcessQ            0          0             0x0

iss:  613440766  snduna:  613440973  sndnxt:  613440973
irs: 2373850036  rcvnxt: 2373850220

sndwnd:  16178  scale:      0  maxrcvwnd:  16384
rcvwnd:  16201  scale:      0  delrcvwnd:    183

SRTT: 607 ms, RTTO: 2949 ms, RTV: 2342 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 1000 ms, ACK hold: 200 ms
uptime: 129454 ms, Sent idletime: 31475 ms, Receive idletime: 31273 ms
Status Flags: passive open, gen tcbs
Option Flags: nagle, path mtu capable
IP Precedence value : 6

Datagrams (max data segment is 1460 bytes):
Rcvd: 12 (out of order: 0), with data: 7, total data bytes: 183
Sent: 14 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with data:
 8, total data bytes: 206

 Packets received in fast path: 0, fast processed: 0, slow path: 0
fast lock acquisition failures: 0, slow path: 0
TCP Semaphore      0x7FB17EDD3048  FREE
```

Configuration Examples for Connecting to a Service Provider Using External BGP

Example: Influencing Inbound Path Selection

The following example shows how you can use route maps to modify incoming data from a neighbor. Any route received from 10.222.1.1 that matches the filter parameters set in autonomous system access list 200 will have its weight set to 200 and its local preference set to 250, and it will be accepted.

```
router bgp 100
```

```
!
 neighbor 10.222.1.1 route-map FIX-WEIGHT in
 neighbor 10.222.1.1 remote-as 1
!
ip as-path access-list 200 permit ^690$
ip as-path access-list 200 permit ^1800
!
route-map FIX-WEIGHT permit 10
 match as-path 200
 set local-preference 250
 set weight 200
```

In the following example, the route map named FINANCE marks all paths originating from autonomous system 690 with an MED metric attribute of 127. The second permit clause is required so that routes not matching autonomous system path list 1 will still be sent to neighbor 10.1.1.1.

```
router bgp 65000
 neighbor 10.1.1.1 route-map FINANCE out
!
ip as-path access-list 1 permit ^690_
ip as-path access-list 2 permit .*
!
route-map FINANCE permit 10
 match as-path 1
 set metric 127
!
route-map FINANCE permit 20
 match as-path 2
```

Inbound route maps could perform prefix-based matching and set various parameters of the update. Inbound prefix matching is available in addition to autonomous system path and community list matching. The following example shows how the route map named SET-LOCAL-PREF sets the local preference of the inbound prefix 172.20.0.0/16 to 120:

```
!
router bgp 65100
 network 10.108.0.0
 neighbor 10.108.1.1 remote-as 65200
 neighbor 10.108.1.1 route-map SET-LOCAL-PREF in
!
route-map SET-LOCAL-PREF permit 10
 match ip address 2
 set local-preference 120
!
route-map SET-LOCAL-PREF permit 20
!
access-list 2 permit 172.20.0.0 0.0.255.255
access-list 2 deny any
```

Example: Influencing Inbound Path Selection by Modifying the AS-path Attribute Using 4-Byte AS Numbers

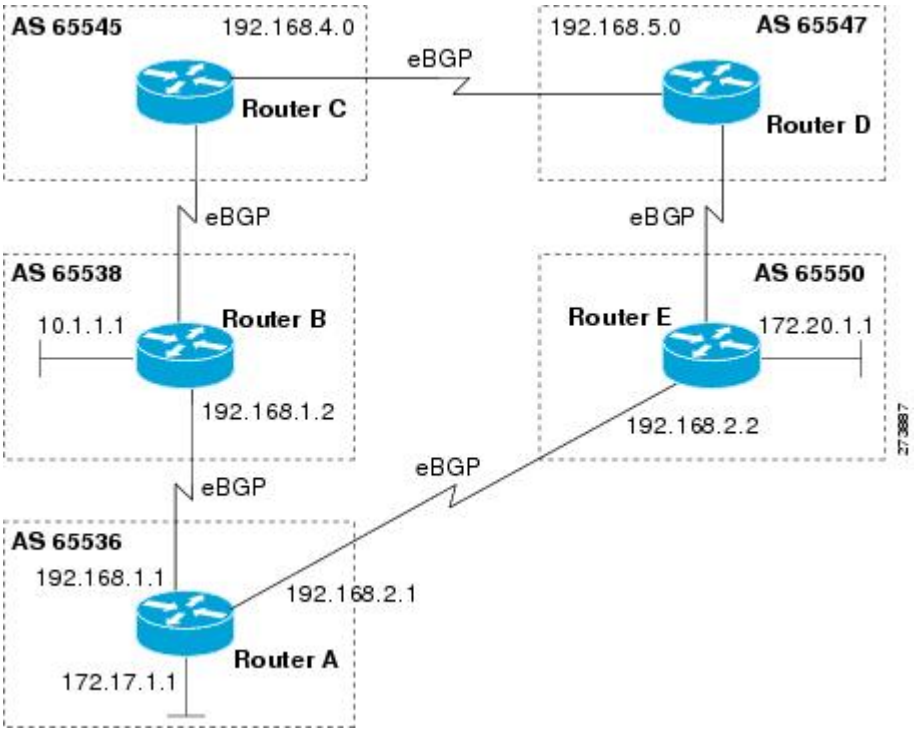
This example shows how to configure BGP to influence the inbound path selection for traffic destined for the 172.17.1.0 network by modifying the AS-path attribute. In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SXI1, and later releases, BGP support for 4-octet (4-byte) autonomous system numbers was introduced. The 4-byte autonomous system numbers in this example are formatted in the default asplain (decimal value) format; for example, Router B is in autonomous system number 65538 in the figure below. For more details

about the introduction of 4-byte autonomous system numbers, see the "BGP Autonomous System Number Formats" section.

One of the methods that BGP can use to influence the choice of paths in another autonomous system is to modify the AS-path attribute. For example, in the figure below, Router A advertises its own network, 172.17.1.0, to its BGP peers in autonomous system 65538 and autonomous system 65550. When the routing information is propagated to autonomous system 65545, the routers in autonomous system 65545 have network reachability information about network 172.17.1.0 from two different routes. The first route is from autonomous system 65538 with an AS-path consisting of 65538, 65536. The second route is through autonomous system 65547 with an AS-path of 65547, 65550, 65536. If all other BGP attribute values are the same, Router C in autonomous system 65545 would choose the route through autonomous system 65538 for traffic destined for network 172.17.1.0 because it is the shortest route in terms of autonomous systems traversed.

Autonomous system 65536 now receives all traffic from autonomous system 65545 for the 172.17.1.0 network through Router B in autonomous system 65538. If, however, the link between autonomous system 65538 and autonomous system 65536 is a really slow and congested link, the **set as-path prepend**command can be used at Router A to influence inbound path selection for the 172.17.1.0 network by making the route through autonomous system 65538 appear to be longer than the path through autonomous system 65550. The configuration is done at Router A in the figure below by applying a route map to the outbound BGP updates to Router B. Using the **set as-path prepend**command, all the outbound BGP updates from Router A to Router B will have their AS-path attribute modified to add the local autonomous system number 65536 twice. After the configuration, autonomous system 65545 receives updates about the 172.17.1.0 network through autonomous system 65538. The new AS-path is 65538, 65536, 65536, 65536, which is now longer than the AS-path from autonomous system 65547 (unchanged at a value of 65547, 65550, 65536). Networking devices in autonomous system 65545 will now prefer the route through autonomous system 65547 to forward packets with a destination address in the 172.17.1.0 network.

Figure 63: Network Topology for Modifying the AS-path Attribute



The configuration for this example is performed at Router A in the figure above.

```
router bgp 65536
 address-family ipv4 unicast
  network 172.17.1.0 mask 255.255.255.0
  neighbor 192.168.1.2 remote-as 65538
  neighbor 192.168.1.2 activate
  neighbor 192.168.1.2 route-map PREPEND out
  exit-address-family
 exit
route-map PREPEND permit 10
set as-path prepend 65536 65536
```

Example: Filtering BGP Prefixes with Prefix Lists

This section contains the following examples:

Example: Filtering BGP Prefixes Using a Single Prefix List

The following example shows how a prefix list denies the default route 0.0.0.0/0:

```
ip prefix-list abc deny 0.0.0.0/0
```

The following example shows how a prefix list permits a route that matches the prefix 10.0.0.0/8:

```
ip prefix-list abc permit 10.0.0.0/8
```

The following example shows how to configure the BGP process so that it accepts only prefixes with a prefix length of /8 to /24:

```
router bgp 40000
 network 10.20.20.0
 distribute-list prefix max24 in
!
ip prefix-list max24 seq 5 permit 0.0.0.0/0 ge 8 le 24
```

The following example configuration shows how to conditionally originate a default route (0.0.0.0/0) in RIP when a prefix 10.1.1.0/24 exists in the routing table:

```
ip prefix-list cond permit 10.1.1.0/24
!
route-map default-condition permit 10
 match ip address prefix-list cond
!
router rip
 default-information originate route-map default-condition
```

The following example shows how to configure BGP to accept routing updates from 192.168.1.1 only, besides filtering on the prefix length:

```
router bgp 40000
 distribute-list prefix max24 gateway allowlist in
!
ip prefix-list allowlist seq 5 permit 192.168.1.1/32
!
```

The following example shows how to direct the BGP process to filter incoming updates to the prefix using name1, and match the gateway (next hop) of the prefix being updated to the prefix list name2, on Gigabit Ethernet interface 0/0/0:

```
router bgp 103
 distribute-list prefix name1 gateway name2 in gigabitethernet 0/0/0
```

Example: Filtering BGP Prefixes Using a Group of Prefixes

The following example shows how to configure BGP to permit routes with a prefix length up to 24 in network 192/8:

```
ip prefix-list abc permit 192.0.0.0/8 le 24
```

The following example shows how to configure BGP to deny routes with a prefix length greater than 25 in 192/8:

```
ip prefix-list abc deny 192.0.0.0/8 ge 25
```

The following example shows how to configure BGP to permit routes with a prefix length greater than 8 and less than 24 in all address space:

```
ip prefix-list abc permit 0.0.0.0/0 ge 8 le 24
```

The following example shows how to configure BGP to deny routes with a prefix length greater than 25 in all address space:

```
ip prefix-list abc deny 0.0.0.0/0 ge 25
```

The following example shows how to configure BGP to deny all routes in network 10/8, because any route in the Class A network 10.0.0.0/8 is denied if its mask is less than or equal to 32 bits:

```
ip prefix-list abc deny 10.0.0.0/8 le 32
```

The following example shows how to configure BGP to deny routes with a mask greater than 25 in 192.168.1.0/24:

```
ip prefix-list abc deny 192.168.1.0/24 ge 25
```

The following example shows how to configure BGP to permit all routes:

```
ip prefix-list abc permit 0.0.0.0/0 le 32
```

Example: Adding or Deleting Prefix List Entries

You can add or delete individual entries in a prefix list if a prefix list has the following initial configuration:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 192.168.0.0/15
```

The following example shows how to delete an entry from the prefix list so that 192.168.0.0 is not permitted, and add a new entry that permits 10.0.0.0/8:

```
no ip prefix-list abc permit 192.168.0.0/15
ip prefix-list abc permit 10.0.0.0/8
```

The new configuration is as follows:

```
ip prefix-list abc deny 0.0.0.0/0 le 7
ip prefix-list abc deny 0.0.0.0/0 ge 25
ip prefix-list abc permit 10.0.0.0/8
```

Example: Filtering Traffic Using COMMUNITIES Attributes

This section contains two examples of the use of BGP COMMUNITIES attributes with route maps.

The first example configures a route map named *set-community*, which is applied to the outbound updates to the neighbor 172.16.232.50. The routes that pass access list 1 are given the well-known COMMUNITIES attribute value **no-export**. The remaining routes are advertised normally. The **no-export** community value automatically prevents the advertisement of those routes by the BGP speakers in autonomous system 200.

```
router bgp 100
 neighbor 172.16.232.50 remote-as 200
 neighbor 172.16.232.50 send-community
 neighbor 172.16.232.50 route-map set-community out
!
route-map set-community permit 10
 match address 1
 set community no-export
!
route-map set-community permit 20
 match address 2
```

The second example configures a route map named *set-community*, which is applied to the outbound updates to neighbor 172.16.232.90. All the routes that originate from autonomous system 70 have the COMMUNITIES attribute values 200 200 added to their already existing communities. All other routes are advertised as normal.

```
route-map bgp 200
 neighbor 172.16.232.90 remote-as 100
 neighbor 172.16.232.90 send-community
 neighbor 172.16.232.90 route-map set-community out
!
route-map set-community permit 10
 match as-path 1
 set community 200 200 additive
!
route-map set-community permit 20
!
ip as-path access-list 1 permit 70$
ip as-path access-list 2 permit .*
```

Example: Filtering Traffic Using AS-Path Filters

The following example shows BGP path filtering by neighbor. Only the routes that pass autonomous system path access list 2 will be sent to 192.168.12.10. Similarly, only routes passing access list 3 will be accepted from 192.168.12.10.

```
router bgp 200
 neighbor 192.168.12.10 remote-as 100
 neighbor 192.168.12.10 filter-list 1 out
 neighbor 192.168.12.10 filter-list 2 in
 exit
ip as-path access-list 1 permit _109_
ip as-path access-list 2 permit _200$
```

```
ip as-path access-list 2 permit ^100$
ip as-path access-list 3 deny _690$
ip as-path access-list 3 permit .*
```

Example: Filtering Traffic with AS-path Filters Using 4-Byte Autonomous System Numbers

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXI1, and Later Releases

The following example is available in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases and shows BGP path filtering by neighbor using 4-byte autonomous system numbers in asplain format. Only the routes that pass autonomous system path access list 2 will be sent to 192.168.3.2.

```
ip as-path access-list 2 permit ^65536$
router bgp 65538
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 65550
  neighbor 192.168.3.2 activate
  neighbor 192.168.3.2 filter-list 2 in
  end
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example available in Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases shows BGP path filtering by neighbor using 4-byte autonomous system numbers in asdot format. Only the routes that pass autonomous system path access list 2 will be sent to 192.168.3.2.

Note In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

```
ip as-path access-list 2 permit ^1\.0$
router bgp 1.2
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 1.14
  neighbor 192.168.3.2 filter-list 2 in
  end
```

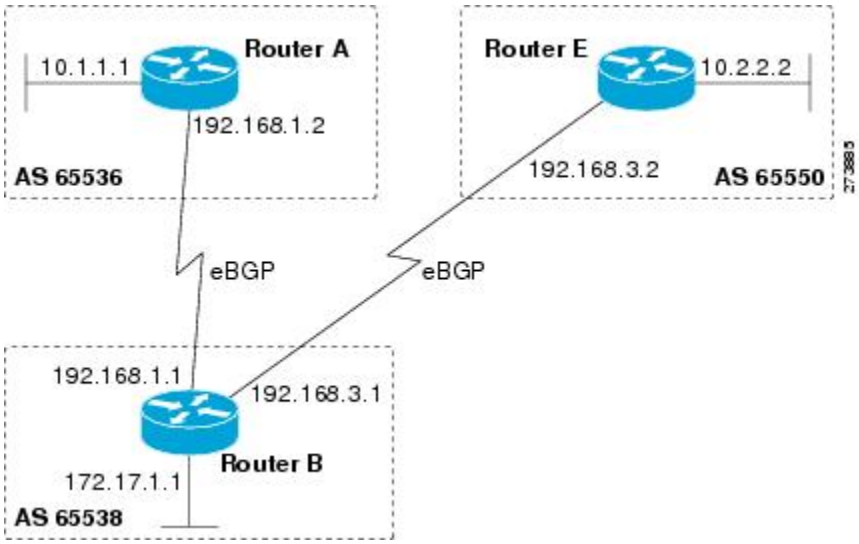
Example: Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers

Asplain Default Format in Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)SXI1, and Later Releases

The following example shows how to filter traffic by creating an extended BGP community list to control outbound routes. In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, and later releases, extended BGP communities support 4-byte autonomous system numbers in the regular

expressions in asplain by default. Extended community attributes are used to configure, filter, and identify routes for VRF instances and MPLS VPNs. The **ip extcommunity-list**command is used to configure named or numbered extended community lists. All of the standard rules of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers.

Figure 64: BGP Topology for Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers in Asplain Format



Note A sequence number is applied to all extended community list entries by default regardless of the configuration mode. Explicit sequencing and resequencing of extended community list entries can be configured only in IP extended community-list configuration mode and not in global configuration mode.

In this exam the figure above is configured with an extended named community list to specify that the BGP peer at 192.1681.2 is not sent advertisements about any path through or from the 4-byte autonomous system 65550. The IP extended community-list configuration mode is used, and the ability to resequence entries is shown.

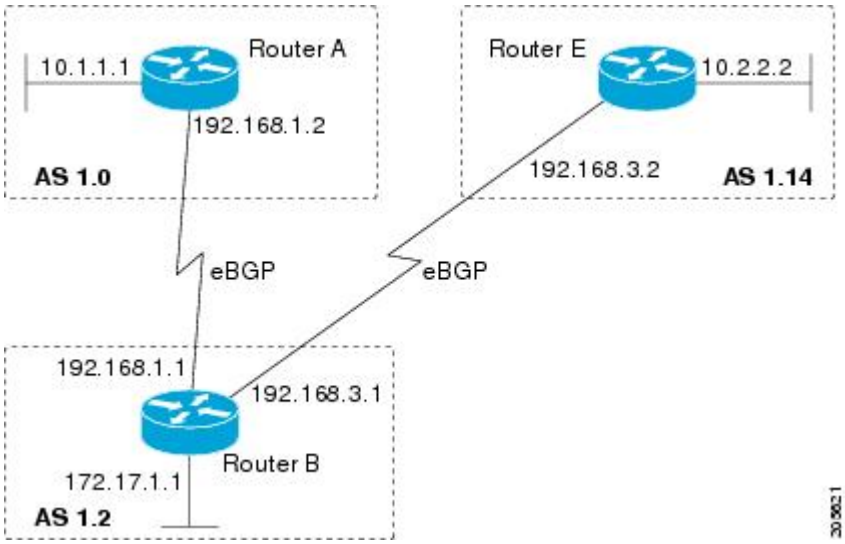
```
ip extcommunity-list expanded DENY65550
 10 deny _65550_
 20 deny ^65550 .*
 resequence 50 100
 exit
router bgp 65538
 network 172.17.1.0 mask 255.255.255.0
 address-family ipv4 unicast
  neighbor 192.168.3.2 remote-as 65550
  neighbor 192.168.1.2 remote-as 65536
  neighbor 192.168.3.2 activate
  neighbor 192.168.1.2 activate
  end
show ip extcommunity-list DENY65550
```

Asdot Default Format in Cisco IOS Release 12.0(32)S12, and 12.4(24)T

The following example shows how to filter traffic by creating an extended BGP community list to control outbound routes. In Cisco IOS Release 12.0(32)S12, 12.4(24)T, and later releases, extended BGP communities support 4-byte autonomous system numbers in the regular expressions in asdot format only. Extended community attributes are used to configure, filter, and identify routes for VRF instances and MPLS VPNs. The **ip extcommunity-list**command is used to configure named or numbered extended community lists. All of the standard rules of access lists apply to the configuration of extended community lists. Regular expressions are supported by the expanded range of extended community list numbers.

Note In Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SXI1, and later releases, this example works if you have configured asdot as the default display format using the **bgp asnotation dot** command.

Figure 65: BGP Topology for Filtering Traffic Using Extended Community Lists with 4-Byte Autonomous System Numbers in Asdot Format



Note A sequence number is applied to all extended community list entries by default regardless of the configuration mode. Explicit sequencing and resequencing of extended community list entries can be configured only in IP extended community-list configuration mode and not in global configuration mode.

In this exam the figure above is configured with an extended named community list to specify that the BGP peer at 192.1681.2 is not sent advertisements about any path through or from the 4-byte autonomous system 65550. The IP extended community-list configuration mode is used, and the ability to resequence entries is shown.

```
ip extcommunity-list expanded DENY114
 10 deny _1\.14_
 20 deny ^1\.14 .*
 resequence 50 100
 exit
router bgp 1.2
 network 172.17.1.0 mask 255.255.255.0
 address-family ipv4 unicast
```

```
  neighbor 192.168.3.2 remote-as 1.14
  neighbor 192.168.1.2 remote-as 1.0
  neighbor 192.168.3.2 activate
  neighbor 192.168.1.2 activate
  end
show ip extcommunity-list DENY114
```

Example: Filtering Traffic Using a BGP Route Map

The following example shows how to use an address family to configure BGP so that any unicast and multicast routes from neighbor 10.1.1.1 are accepted if they match access list 1:

```
route-map filter-some-multicast
 match ip address 1
 exit
router bgp 65538
 neighbor 10.1.1.1 remote-as 65537
 address-family ipv4 unicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-map filter-some-multicast in
  exit
 exit
router bgp 65538
 neighbor 10.1.1.1 remote-as 65537
 address-family ipv4 multicast
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-map filter-some-multicast in
  end
```

Where to Go Next

- To configure advanced BGP feature tasks, proceed to the “Configuring Advanced BGP Features” module.
- To configure BGP neighbor session options, proceed to the “Configuring BGP Neighbor Session Options” module.
- To configure internal BGP tasks, proceed to the “Configuring Internal BGP Features” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
BGP overview	“Cisco BGP Overview” module

Related Topic	Document Title
Configuring basic BGP tasks	"Configuring a Basic BGP Network" module
BGP fundamentals and description	*Large-Scale IP Network Solutions*, Khalid Raza and Mark Turner, Cisco Press, 2000
Implementing and controlling BGP in scalable networks	*Building Scalable Cisco Networks*, Catherine Paquet and Diane Teare, Cisco Press, 2001
Interdomain routing basics	*Internet Routing Architectures*, Bassam Halabi, Cisco Press, 1997

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2519	*A Framework for Inter-Domain Route Aggregation*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 4271	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4684	*Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC	Title
RFC 4893	*BGP Support for Four-Octet AS Number Space*
RFC 5291	*Outbound Route Filtering Capability for BGP-4*
RFC 5396	*Textual Representation of Autonomous system (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*
RFC 8212	Default External BGP (EBGP) Route Propagation Behavior without Policies

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Connecting to a Service Provider Using External BGP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

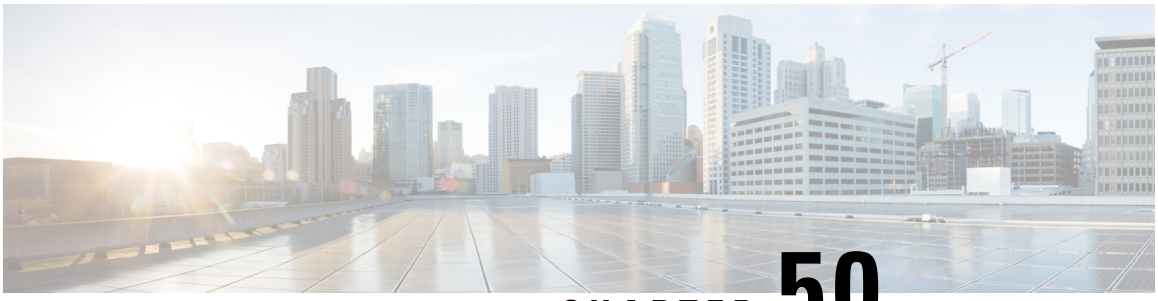
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 75: Feature Information for Connecting to a Service Provider Using External BGP

Feature Name	Releases	Feature Configuration Information
BGP Increased Support of Numbered AS-Path Access Lists to 500	12.0(22)S 12.2(15)T 12.2(18)S 12.2(18)SXD 12.2(27)SBC 15.0(1)S	The BGP Increased Support of Numbered AS-Path Access Lists to 500 feature increases the maximum number of autonomous systems access lists that can be configured using the **ip as-path access-list** command from 199 to 500.

Feature Name	Releases	Feature Configuration Information
BGP Named Community Lists	12.2(8)T 12.2(14)S 15.0(1)S	The BGP Named Community Lists feature introduces a new type of community list called the named community list. The BGP Named Community Lists feature allows the network operator to assign meaningful names to community lists and increases the number of community lists that can be configured. A named community list can be configured with regular expressions and with numbered community lists. All rules of numbered communities apply to named community lists except that there is no limitation on the number of community attributes that can be configured for a named community list.
BGP Route-Map Policy List Support	12.0(22)S 12.2(15)T 12.2(18)S 12.2(18)SXD 12.2(27)SBC 15.0(1)S	The BGP Route-Map Policy List Support feature introduces new functionality to BGP route maps. This feature adds the capability for a network operator to group route map match clauses into named lists called policy lists. A policy list functions like a macro. When a policy list is referenced in a route map, all of the match clauses are evaluated and processed as if they had been configured directly in the route map. This enhancement simplifies the configuration of BGP routing policy in medium-size and large networks because a network operator can preconfigure policy lists with groups of match clauses and then reference these policy lists within different route maps. The network operator no longer needs to manually reconfigure each recurring group of match clauses that occur in multiple route map entries.
BGP Support for Named Extended Community Lists	12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(11)T 15.0(1)S	The BGP Support for Named Extended Community Lists feature introduces the ability to configure extended community lists using names in addition to the existing numbered format.
BGP Support for Sequenced Entries in Extended Community Lists	12.2(25)S 12.2(27)SBC 12.2(33)SRA 12.2(33)SXH 12.3(11)T 15.0(1)S	The BGP Support for Sequenced Entries in Extended Community Lists feature introduces automatic sequencing of individual entries in BGP extended community lists. This feature also introduces the ability to remove or resequence extended community list entries without deleting the entire existing extended community list.
BGP 4 Prefix Filter and Inbound Route Maps	Cisco IOS XE 3.1.0SG	

Feature Name	Releases	Feature Configuration Information
EBGP Route Propagation without Policies	Cisco IOS XE Amsterdam 17.2.1	By default, an External BGP (EBGP) router propagates routes to and from an EBGP neighbor when you have not configured inbound and outbound policies . From Cisco IOS XE Release 17.2.1, you can modify this default behavior. You can configure an EBGP router not to propagate routes to and from a neighbor unless you configure at least one inbound and one outbound policy for the neighbor. The following commands were introduced or modified by this feature: **bgp safe-ebgp-policy**, **show ip bgp** *address-family* neighbor *ip-address*, **show ip bgp** *address-family* **summary**.



CHAPTER 50

BGP Route-Map Continue

The BGP Route-Map Continue feature introduces the continue clause to BGP route-map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configuration need not be repeated within the same route map.

Information About BGP Route Map Continue

BGP Route Map with a Continue Clause

In BGP route-map configuration, the continue clause allows for more programmable policy configuration and route filtering and introduced the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow you to configure and organize more modular policy definitions so that specific policy configurations need not be repeated within the same route map. Before the continue clause was introduced, route-map configuration was linear and did not allow any control over the flow of a route map.

Route Map Operation Without Continue Clauses

A route map evaluates match clauses until a successful match occurs. After the match occurs, the route map stops evaluating match clauses and starts executing set clauses, in the order in which they were configured. If a successful match does not occur, the route map "falls through" and evaluates the next sequence number of the route map until all configured route map entries have been evaluated or a successful match occurs. Each route map sequence is tagged with a sequence number to identify the entry. Route map entries are evaluated in order starting with the lowest sequence number and ending with the highest sequence number. If the route map contains only set clauses, the set clauses will be executed automatically, and the route map will not evaluate any other route map entries.

Route Map Operation with Continue Clauses

When a continue clause is configured, the route map will continue to evaluate and execute match clauses in the specified route map entry after a successful match occurs. The continue clause can be configured to go to (jump to) a specific route map entry by specifying the sequence number, or if a sequence number is not specified, the continue clause will go to the next sequence number. This behavior is called an "implied continue." If a match clause exists, the continue clause is executed only if a match occurs. If no successful matches occur, the continue clause is ignored.

Match Operations with Continue Clauses

If a match clause does not exist in the route map entry but a continue clause does, the continue clause will be automatically executed and go to the specified route map entry. If a match clause exists in a route map entry, the continue clause is executed only when a successful match occurs. When a successful match occurs and a continue clause exists, the route map executes the set clauses and then goes to the specified route map entry. If the next route map entry contains a continue clause, the route map will execute the continue clause if a successful match occurs. If a continue clause does not exist in the next route map entry, the route map will be evaluated normally. If a continue clause exists in the next route map entry but a match does not occur, the route map will not continue and will "fall through" to the next sequence number if one exists.

Note If the number of community lists in a match community clause within a route map exceed 256 characters in a line, you must nvgen multiple match community statements in a new line.

Set Operations with Continue Clauses

Set clauses are saved during the match clause evaluation process and are executed after the route-map evaluation is completed. The set clauses are evaluated and executed in the order in which they were configured. Set clauses are executed only after a successful match occurs, unless the route map does not contain a match clause. The continue statement proceeds to the specified route map entry only after configured set actions are performed. If a set action occurs in the first route map and then the same set action occurs again, with a different value, in a subsequent route map entry, the last set action may override any previous set actions that were configured with the same **set** command unless the **set** command permits more than one value. For example, the **set as-path prepend** command permits more than one autonomous system number to be configured.

Note A continue clause can be executed, without a successful match, if a route map entry does not contain a match clause.

Note Route maps have a linear behavior, not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. For an example, see the "Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map" section.

How to Filter Traffic Using Continue Clauses in a BGP Route Map

Filtering Traffic Using Continue Clauses in a BGP Route Map

Perform this task to filter traffic using continue clauses in a BGP route map.

Note Continue clauses can go only to a higher route map entry (a route map entry with a higher sequence number) and cannot go to a lower route map entry.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
5. **neighbor** {*ip-address*| *peer-group-name*} **route-map** *map-name* {**in** | **out**}
6. **exit**
7. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
8. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
9. **set community** *community-number* [**additive**] [*well-known-community*] | **none**}
10. **continue** [*sequence-number*]
11. **end**
12. **show route-map** [*map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode, and creates a BGP routing process.

	Command or Action	Purpose
Step 4	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.0.0.1 remote-as 50000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**neighbor** {*ip-address*| *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in`	Applies the inbound route map to routes received from the specified neighbor, or applies an outbound route map to routes advertised to the specified neighbor.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 7	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME permit 10`	Enters route-map configuration mode to create or configure a route map.
Step 8	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Device(config-route-map)# match ip address 1`	Configures a **match** command that specifies the conditions under which policy routing and route filtering occur. • Multiple **match** commands can be configured. If a **match** command is configured, a match must occur in order for the continue statement to be executed. If a **match** command is not configured, set and continue clauses will be executed. **Note** The **match** and **set** commands used in this task are examples that are used to help describe the operation of the **continue** command. For a list of specific **match** and **set** commands, see the **continue** command in the *Cisco IOS IP Routing: BGP Command Reference*.
Step 9	**set community** *community-number* [**additive**] [*well-known-community*] | **none**} **Example:** `Device(config-route-map)# set community 10:1`	Configures a **set** command that specifies the routing action to perform if the criteria enforced by the **match** commands are met. • Multiple **set** commands can be configured. • In this example, a clause is created to set the specified community.

	Command or Action	Purpose
Step 10	**continue** [*sequence-number*] **Example:** `Device(config-route-map)# continue`	Configures a route map to continue to evaluate and execute match statements after a successful match occurs. • If a sequence number is configured, the continue clause will go to the route map with the specified sequence number. • If no sequence number is specified, the continue clause will go to the route map with the next sequence number. This behavior is called an "implied continue." **Note** Continue clauses in outbound route maps are supported in Cisco IOS XE Release 2.1 and later releases.
Step 11	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.
Step 12	**show route-map** [*map-name*] **Example:** `Device# show route-map`	(Optional) Displays locally configured route maps. The name of the route map can be specified in the syntax of this command to filter the output.

Examples

The following sample output shows how to verify the configuration of continue clauses using the **show route-map** command. The output displays configured route maps including the match, set, and continue clauses.

```
Device# show route-map

route-map MARKETING, permit, sequence 10
  Match clauses:
    ip address (access-lists): 1
    metric 10
  Continue: sequence 40
  Set clauses:
    as-path prepend 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 20
  Match clauses:
    ip address (access-lists): 2
    metric 20
  Set clauses:
    as-path prepend 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 30
  Match clauses:
  Continue: to next entry 40
```

```
  Set clauses:
    as-path prepend 10 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 40
  Match clauses:
    community (community-list filter): 10:1
  Set clauses:
    local-preference 104
  Policy routing matches: 0 packets, 0 bytes
route-map MKTG-POLICY-MAP, permit, sequence 10
  Match clauses:
  Set clauses:
    community 655370
  Policy routing matches: 0 packets, 0 bytes
```

Configuration Examples for BGP Route Map Continue

Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map

The following example shows continue clause configuration in a route map sequence.

Note Continue clauses in outbound route maps are supported only in Cisco IOS Release 12.0(31)S, 12.2(33)SB, 12.2(33)SRB, 12.2(33)SXI, 12.4(4)T, and later releases.

The first continue clause in route map entry 10 indicates that the route map will go to route map entry 30 if a successful matches occurs. If a match does not occur, the route map will "fall through" to route map entry 20. If a successful match occurs in route map entry 20, the set action will be executed and the route map will not evaluate any additional route map entries. Only the first successful **match ip address** clause is supported.

If a successful match does not occur in route map entry 20, the route map will fall through to route map entry 30. This sequence does not contain a match clause, so the set clause will be automatically executed and the **continue** clause will go to the next route map entry because a sequence number is not specified.

If there are no successful matches, the route map will fall through to route map entry 30 and execute the set clause. A sequence number is not specified for the **continue** clause, so route map entry 40 will be evaluated.

There are two behaviors that can occur when the same **set** command is repeated in subsequent **continue** clause entries. For **set** commands that configure an additive or accumulative value (for example, **set community additive**, **set extended community additive**, and **set as-path prepend**), subsequent values are added by subsequent entries. The following example illustrates this behavior. After each set of match clauses, a **set as-path prepend** command is configured to add an autonomous system number to the as-path. After a match occurs, the route map stops evaluating match clauses and starts executing the set clauses, in the order in which they were configured. Depending on how many successful match clauses occur, the as-path is prepended by one, two, or three autonomous system numbers.

```
route-map ROUTE-MAP-NAME permit 10
 match ip address 1
 match metric 10
 set as-path prepend 10
 continue 30
 !
route-map ROUTE-MAP-NAME permit 20
```

```
 match ip address 2
 match metric 20
 set as-path prepend 10 10
 !
route-map ROUTE-MAP-NAME permit 30
 set as-path prepend 10 10 10
 continue
 !
route-map ROUTE-MAP-NAME permit 40
 match community 10:1
 set local-preference 104
```

In this example, the same **set** command is repeated in subsequent **continue** clause entries, but the behavior is different from the first example. For **set** commands that configure an absolute value, the value from the last instance will overwrite the previous value(s). The following example illustrates this behavior. The set clause value in sequence 20 overwrites the set clause value from sequence 10. The next hop for prefixes from the 172.16/16 network is set to 10.2.2.2, not 10.1.1.1.

```
ip prefix-list 1 permit 172.16.0.0/16
ip prefix-list 2 permit 192.168.1.0/24
route-map RED permit 10
 match ip address prefix-list 1
 set ip next hop 10.1.1.1
 continue 20
 exit
route-map RED permit 20
 match ip address prefix-list 2
 set ip next hop 10.2.2.2
 end
```

Note Route maps have a linear behavior and not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. The following example illustrates this case.

In the following example, when routes match an as-path of 10, 20, or 30, the routes are permitted and the continue clause jumps over the explicit deny clause to process the match ip address prefix list. If a match occurs here, the route metric is set to 100. Only routes that do not match an as-path of 10, 20, or 30 and do match a community number of 30 are denied. To deny other routes, you must configure an explicit deny statement.

```
route-map test permit 10
 match as-path 10 20 30
 continue 30
 exit
route-map test deny 20
 match community 30
 exit
route-map test permit 30
 match ip address prefix-list 1
 set metric 100
 exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

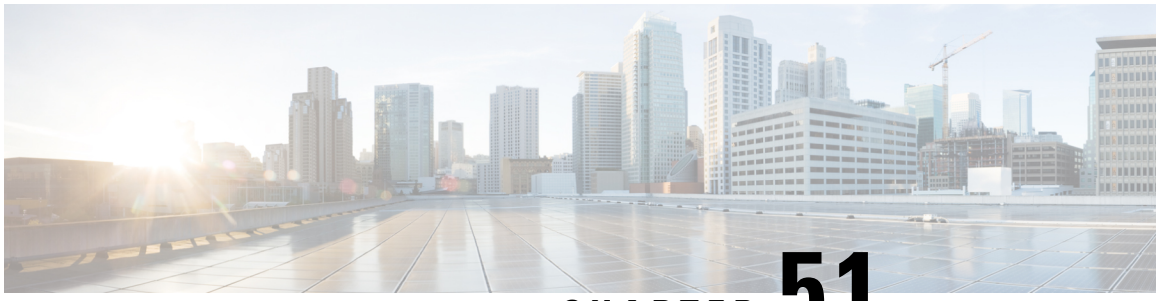
Feature Information for BGP Route Map Continue

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 76: Feature Information for BGP Route Map Continue

Feature Name	Releases	Feature Information
BGP Route Map Continue		The BGP Route Map Continue feature introduces the continue clause to BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering and introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow the network operator to configure and organize more modular policy definitions so that specific policy configuration need not be repeated within the same route map.



CHAPTER 51

BGP Route-Map Continue Support for Outbound Policy

The BGP Route-Map Continue Support for an Outbound Policy feature introduces support for continue clauses to be applied to outbound route maps.

Iinformation About BGP Route-Map Continue Support for Outbound Policy

BGP Route Map with a Continue Clause

Subsequent to the Cisco implementation of route maps, the continue clause was introduced into BGP route map configuration. The continue clause allows for more programmable policy configuration and route filtering. The continue clause introduces the ability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow you to configure and organize more modular policy definitions so that specific policy configurations need not be repeated within the same route map. Before the continue clause was introduced, route map configuration was linear and did not allow any control over the flow of a route map.

Route Map Operation Without Continue Clauses

A route map evaluates match clauses until a successful match occurs. After the match occurs, the route map stops evaluating match clauses and starts executing set clauses, in the order in which they were configured. If a successful match does not occur, the route map "falls through" and evaluates the next sequence number of the route map until all configured route map entries have been evaluated or a successful match occurs. Each route map sequence is tagged with a sequence number to identify the entry. Route map entries are evaluated in order starting with the lowest sequence number and ending with the highest sequence number. If the route map contains only set clauses, the set clauses will be executed automatically, and the route map will not evaluate any other route map entries.

Route Map Operation with Continue Clauses

When a continue clause is configured, the route map will continue to evaluate and execute match clauses in the specified route map entry after a successful match occurs. The continue clause can be configured to go to (jump to) a specific route map entry by specifying the sequence number, or if a sequence number is not specified, the continue clause will go to the next sequence number. This behavior is called an "implied continue." If a match clause exists, the continue clause is executed only if a match occurs. If no successful matches occur, the continue clause is ignored.

Match Operations with Continue Clauses

If a match clause does not exist in the route map entry but a continue clause does, the continue clause will be automatically executed and go to the specified route map entry. If a match clause exists in a route map entry, the continue clause is executed only when a successful match occurs. When a successful match occurs and a continue clause exists, the route map executes the set clauses and then goes to the specified route map entry. If the next route map entry contains a continue clause, the route map will execute the continue clause if a successful match occurs. If a continue clause does not exist in the next route map entry, the route map will be evaluated normally. If a continue clause exists in the next route map entry but a match does not occur, the route map will not continue and will "fall through" to the next sequence number if one exists.

Note If the number of community lists in a match community clause within a route map exceed 256 characters in a line, you must nvgen multiple match community statements in a new line.

Set Operations with Continue Clauses

Set clauses are saved during the match clause evaluation process and are executed after the route-map evaluation is completed. The set clauses are evaluated and executed in the order in which they were configured. Set clauses are executed only after a successful match occurs, unless the route map does not contain a match clause. The continue statement proceeds to the specified route map entry only after configured set actions are performed. If a set action occurs in the first route map and then the same set action occurs again, with a different value, in a subsequent route map entry, the last set action may override any previous set actions that were configured with the same **set** command unless the **set** command permits more than one value. For example, the **set as-path prepend** command permits more than one autonomous system number to be configured.

Note A continue clause can be executed, without a successful match, if a route map entry does not contain a match clause.

Note Route maps have a linear behavior, not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route-map. For an example, see the "Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map" section.

How to Filter Traffic Using Continue Clauses in a BGP Route Map

Filtering Traffic Using Continue Clauses in a BGP Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
6. **neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
10. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
11. **set community** { { [*community-number*] [*well-known-community*] [**additive**]} | **none**}
12. **continue** [*sequence-number*]
13. **end**
14. **show route-map** [*map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:**	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.

	Command or Action	Purpose
	`Device(config-router)# neighbor 10.0.0.1 remote-as 50000`	
Step 5	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the device is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router-af)# neighbor 10.0.0.1 route-map ROUTE-MAP-NAME in`	Applies the inbound route map to routes received from the specified neighbor, or applies an outbound route map to routes advertised to the specified neighbor.
Step 7	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 8	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 9	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME permit 10`	Enters route-map configuration mode to create or configure a route map.
Step 10	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Device(config-route-map)# match ip address 1`	Configures a **match** command that specifies the conditions under which policy routing and route filtering occur. • Multiple **match** commands can be configured. If a **match** command is configured, a match must occur in order for the continue statement to be executed. If

	Command or Action	Purpose
		a **match** command is not configured, set and continue clauses will be executed. **Note** The **match** and **set** commands used in this task are examples that are used to help describe the operation of the **continue** command. For a list of specific **match** and **set** commands, see the **continue** command in the *Cisco IOS IP Routing: BGP Command Reference*.
Step 11	**set community** { { [*community-number*] [*well-known-community*] [**additive**]} | **none**} **Example:** `Device(config-route-map)# set community 10:1`	Configures a **set** command that specifies the routing action to perform if the criteria enforced by the **match** commands are met. • Multiple **set** commands can be configured. • In this example, a clause is created to set the specified community number in aa:nn format.
Step 12	**continue** [*sequence-number*] **Example:** `Device(config-route-map)# continue`	Configures a route map to continue to evaluate and execute match statements after a successful match occurs. • If a sequence number is configured, the continue clause will go to the route map with the specified sequence number. • If no sequence number is specified, the continue clause will go to the route map with the next sequence number. This behavior is called an "implied continue."
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.
Step 14	**show route-map** [*map-name*] **Example:** `Device# show route-map`	(Optional) Displays locally configured route maps. The name of the route map can be specified in the syntax of this command to filter the output.

Examples

The following sample output shows how to verify the configuration of continue clauses using the **show route-map** command. The output displays configured route maps including the match, set, and continue clauses.

```
Device# show route-map
```

```
route-map MARKETING, permit, sequence 10
  Match clauses:
    ip address (access-lists): 1
    metric 10
  Continue: sequence 40
  Set clauses:
    as-path prepend 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 20
  Match clauses:
    ip address (access-lists): 2
    metric 20
  Set clauses:
    as-path prepend 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 30
  Match clauses:
  Continue: to next entry 40
  Set clauses:
    as-path prepend 10 10 10
  Policy routing matches: 0 packets, 0 bytes
route-map MARKETING, permit, sequence 40
  Match clauses:
    community (community-list filter): 10:1
  Set clauses:
    local-preference 104
  Policy routing matches: 0 packets, 0 bytes
route-map MKTG-POLICY-MAP, permit, sequence 10
  Match clauses:
  Set clauses:
    community 655370
  Policy routing matches: 0 packets, 0 bytes
```

Configuration Examples for BGP Route-Map Continue Support for Outbound Policy

Examples: Filtering Traffic Using Continue Clauses in a BGP Route Map

The following example shows continue clause configuration in a route map sequence.

The first continue clause in route map entry 10 indicates that the route map will go to route map entry 30 if a successful matches occurs. If a match does not occur, the route map will “fall through” to route map entry 20. If a successful match occurs in route map entry 20, the set action will be executed and the route map will not evaluate any additional route map entries. Only the first successful match ip address clause is supported.

If a successful match does not occur in route map entry 20, the route map will fall through to route map entry 30. This sequence does not contain a match clause, so the set clause will be automatically executed and the continue clause will go to the next route map entry because a sequence number is not specified.

If there are no successful matches, the route map will fall through to route map entry 30 and execute the set clause. A sequence number is not specified for the continue clause, so route map entry 40 will be evaluated.

There are two behaviors that can occur when the same **set** command is repeated in subsequent continue clause entries. For **set** commands that configure an additive or accumulative value (for example, **set community additive**, **set extended community additive**, and **set as-path prepend**), subsequent values are added by

subsequent entries. The following example illustrates this behavior. After each set of match clauses, a **set as-path prepend** command is configured to add an autonomous system number to the as-path. After a match occurs, the route map stops evaluating match clauses and starts executing the set clauses, in the order in which they were configured. Depending on the number of successful match clauses, the as-path is prepended by one, two, or three autonomous system numbers.

```
route-map ROUTE-MAP-NAME permit 10
 match ip address 1
 match metric 10
 set as-path prepend 10
 continue 30
 !
route-map ROUTE-MAP-NAME permit 20
 match ip address 2
 match metric 20
 set as-path prepend 10 10
 !
route-map ROUTE-MAP-NAME permit 30
 set as-path prepend 10 10 10
 continue
 !
route-map ROUTE-MAP-NAME permit 40
 match community 10:1
 set local-preference 104
```

In this example, the same **set** command is repeated in subsequent continue clause entries but the behavior is different from the first example. For **set** commands that configure an absolute value, the value from the last instance will overwrite the previous value(s). The following example illustrates this behavior. The set clause value in sequence 20 overwrites the set clause value from sequence 10. The next hop for prefixes from the 172.16/16 network is set to 10.2.2.2 and not 10.1.1.1.

```
ip prefix-list 1 permit 172.16.0.0/16
ip prefix-list 2 permit 192.168.1.0/24
route-map RED permit 10
 match ip address prefix-list 1
 set ip next hop 10.1.1.1
 continue 20
 exit
route-map RED permit 20
 match ip address prefix-list 2
 set ip next hop 10.2.2.2
 end
```

Note Route maps have a linear behavior, not a nested behavior. Once a route is matched in a route map permit entry with a continue command clause, it will not be processed by the implicit deny at the end of the route map. The following example illustrates this case.

In the following example, when routes match an AS-path of 10, 20, or 30, the routes are permitted and the continue clause jumps over the explicit deny clause to process the **match ip address prefix-list** command. If a match occurs here, the route metric is set to 100. Only routes that do not match an AS-path of 10, 20, or 30 and do match a community number of 30 are denied. To deny other routes, you must configure an explicit deny statement.

```
route-map test permit 10
 match as-path 10 20 30
```

```
 continue 30
 exit
route-map test deny 20
 match community 30
 exit
route-map test permit 30
 match ip address prefix-list 1
 set metric 100
 exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

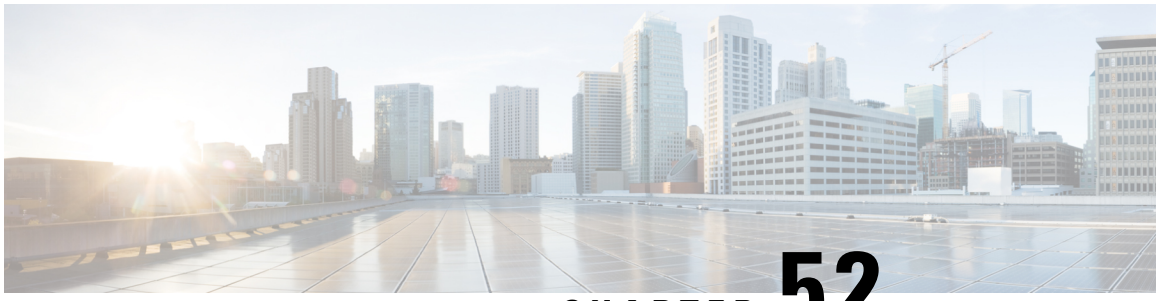
Feature Information for BGP Route-Map Continue Support for Outbound Policy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 77: Feature Information for BGP Route-Map Continue Support for Outbound Policy

Feature Name	Releases	Feature Information
BGP Route-Map Continue Support for Outbound Policy		The BGP Route-Map Continue Support for an Outbound Policy feature introduces support for continue clauses to be applied to outbound route maps.



CHAPTER 52

Removing Private AS Numbers from the AS Path in BGP

Private autonomous system numbers (ASNs) are used by ISPs and customer networks to conserve globally unique AS numbers. Private AS numbers cannot be used to access the global Internet because they are not unique. AS numbers appear in eBGP AS paths in routing updates. Removing private ASNs from the AS path is necessary if you have been using private ASNs and you want to access the global Internet.

Restrictions on Removing and Replacing Private ASNs from the AS Path

- The feature applies to eBGP neighbors only.
- The feature applies to routers in a public AS only. The workaround to this restriction would be to apply the **neighbor local-as** command on a per-neighbor basis, with the local AS number being a public AS number.

Information About Removing and Replacing Private ASNs from the AS Path

Public and Private AS Numbers

Public AS numbers are assigned by InterNIC and are globally unique. They range from 1 to 64511. Private AS numbers are used to conserve globally unique AS numbers, and they range from 64512 to 65535. Private AS numbers cannot be leaked to a global BGP routing table because they are not unique, and BGP best path

calculations require unique AS numbers. Therefore, it might be necessary to remove private AS numbers from an AS path before the routes are propagated to a BGP peer.

Benefit of Removing and Replacing Private ASNs from the AS Path

External BGP requires that globally unique AS numbers be used when routing to the global Internet. Using private AS numbers (which are not unique) would prevent access to the global Internet. This feature allows routers that belong to a private AS to access the global Internet. A network administrator configures the routers to remove private AS numbers from the AS path contained in outgoing update messages and optionally, to replace those numbers with the ASN of the local router, so that the AS Path length remains unchanged.

Former Restrictions to Removing Private ASNs from the AS Path

The ability to remove private AS numbers from the AS path has been available for a long time. Prior to Cisco IOS XE Release 3.1S, this feature had the following restrictions:

- If the AS path included both private and public AS numbers, using the **neighbor remove-private-as** command would not remove the private AS numbers.
- If the AS path contained confederation segments, using the **neighbor remove-private-as** command would remove private AS numbers only if the private AS numbers followed the confederation portion of the autonomous path.
- If the AS path contained the AS number of the eBGP neighbor, the private AS numbers would not be removed.

Enhancements to Removing Private ASNs from the AS Path

The ability to remove and replace private AS numbers from the AS path is enhanced in the following ways:

- The **neighbor remove-private-as** command will remove private AS numbers from the AS path even if the path contains both public and private ASNs.
- The **neighbor remove-private-as** command will remove private AS numbers even if the AS path contains only private AS numbers. There is no likelihood of a 0-length AS path because this command can be applied to eBGP peers only, in which case the AS number of the local router is appended to the AS path.
- The **neighbor remove-private-as** command will remove private AS numbers even if the private ASNs appear before the confederation segments in the AS path.
- The **replace-as** keyword is available to replace the private AS numbers being removed from the path with the local AS number, thereby retaining the same AS path length.
- The feature can be applied to neighbors per address family (address family configuration mode). Therefore, you can apply the feature for a neighbor in one address family and not on another, affecting update messages on the outbound side for only the address family for which the feature is configured.
- The feature can be applied in peer group template mode.
- When the feature is configured, output from the **show ip bgp update-group** and **show ip bgp neighbor** commands indicates that private AS numbers were removed or replaced.

How to Remove and Replace Private ASNs from the AS Path

Removing and Replacing Private ASNs from the AS Path (Cisco IOS XE Release 3.1S and Later)

To remove private AS numbers from the AS path on the outbound side of an eBGP neighbor, perform the following task. To also replace private AS numbers with the local router's AS number, include the **all replace-as** keywords in Step 17.

The examples in this task reflect the configuration for Router 2 in the scenario in the figure below.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **exit**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **exit**
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **exit**
12. **router bgp** *autonomous-system-number*
13. **network** *network-number*
14. **network** *network-number*
15. **neighbor** {*ip-address* | *ipv6-address*[**%**]| *peer-group-name*} **remote-as** *autonomous-system-number*
16. **neighbor** {*ip-address* | *ipv6-address*[**%**]| *peer-group-name*} **remote-as** *autonomous-system-number*
17. **neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** [**all** [**replace-as**]]
18. **end**
19. **show ip bgp update-group**
20. **show ip bgp neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0`	Configures an interface.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.30.1.1 255.255.0.0`	Sets a primary or secondary IP address for an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns to the next highest configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface serial 0/0`	Configures an interface.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.16.0.2 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Returns to the next highest configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface serial 1/0`	Configures an interface.
Step 10	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.0.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 11	**exit** **Example:**	Returns to the next highest configuration mode.

	Command or Action	Purpose
	`Router(config-if)# exit`	
Step 12	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Specifies a BGP instance.
Step 13	**network** *network-number* **Example:** `Router(config-router)# network 172.30.0.0`	Specifies a network to be advertised by BGP.
Step 14	**network** *network-number* **Example:** `Router(config-router)# network 192.168.0.0`	Specifies a network to be advertised by BGP.
Step 15	**neighbor** {*ip-address* | *ipv6-address*[**%**]| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 172.16.0.1 remote-as 65000`	Adds an entry to the routing table. • This example configures Router 3 as an eBGP neighbor in private AS 65000.
Step 16	**neighbor** {*ip-address* | *ipv6-address*[**%**]| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.0.2 remote-as 1`	Adds an entry to the routing table. • This example configures Router 1 as an eBGP neighbor in public AS 1.
Step 17	**neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** [**all** [**replace-as**]] **Example:** `Router(config-router)# neighbor 192.168.0.2 remove-private-as all replace-as`	Removes private AS numbers from the AS Path in outgoing updates. • This example removes the private AS numbers from the AS path in outgoing eBGP updates and replaces them with 5, which is the public AS number of the local router.
Step 18	**end** **Example:** `Router(config-router)# end`	Ends the current configuration mode and returns to privileged EXEC mode.
Step 19	**show ip bgp update-group** **Example:** `Router# show ip bgp update-group`	(Optional) Displays information about BGP update groups.

	Command or Action	Purpose
Step 20	**show ip bgp neighbors** **Example:** `Router# show ip bgp neighbors`	(Optional) Displays information about BGP neighbors.

Configuration Examples for Removing and Replacing Private ASNs from the AS Path

Example Removing Private ASNs (Cisco IOS XE Release 3.1S)

In the example below, Router A has the **neighbor remove-private-as** command configured, which removes private AS numbers in updates sent to the neighbor at 172.30.0.7. The subsequent **show** command asks for information about the route to host 1.1.1.1. The output includes private AS numbers 65200, 65201, 65201 in the AS path of 1001 65200 65201 65201 1002 1003 1003.

To prove that the private AS numbers were removed from the AS path, the **show** command on Router B also asks for information about the route to host 1.1.1.1. The output indicates a shorter AS path of 100 1001 1002 1003 1003, which excludes private AS numbers 65200, 65201, and 65201. The 100 prepended in the path is Router B's own AS number.

Router A

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 19.0.101.1 remote-as 1001
 neighbor 172.30.0.7 remote-as 200
 neighbor 172.30.0.7 remove-private-as all
 no auto-summary

RouterA# show ip bgp 1.1.1.1
BGP routing table entry for 1.1.1.1/32, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     1          2
  1001 65200 65201 65201 1002 1003 1003
    19.0.101.1 from 19.0.101.1 (19.0.101.1)
      Origin IGP, localpref 100, valid, external, best RouterA#
```

Router B (All Private ASNs Have Been Removed)

```
RouterB# show ip bgp 1.1.1.1
BGP routing table entry for 1.1.1.1/32, version 3
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  100 1001 1002 1003 1003
    172.30.0.6 from 172.30.0.6 (19.1.0.1)
      Origin IGP, localpref 100, valid, external, best RouterB#
```

Example Removing and Replacing Private ASNs (Cisco IOS XE Release 3.1S)

In the following example, when Router A sends prefixes to the peer 172.30.0.7, all private ASNs in the AS path are replaced with the router's own ASN, which is 100.

Router A

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 172.16.101.1 remote-as 1001
 neighbor 172.16.101.1 update-source Loopback0
 neighbor 172.30.0.7 remote-as 200
 neighbor 172.30.0.7 remove-private-as all replace-as
 no auto-summary
```

Router A receives 1.1.1.1 from peer 172.16.101.1 which has some private ASNs (65200, 65201, and 65201) in the AS path list, as shown in the following output:

```
RouterA# show ip bgp 1.1.1.1
BGP routing table entry for 1.1.1.1/32, version 2
Paths: (1 available, best #1, table default)
   Advertised to update-groups:
      1          2
   1001 65200 65201 65201 1002 1003 1003
     172.16.101.1 from 172.16.101.1 (172.16.101.1)
       Origin IGP, localpref 100, valid, external, best RouterA#
```

Because Router A is configured with **neighbor 172.30.0.7 remove-private-as all replace-as**, Router A sends prefix 1.1.1.1 with all private ASNs replaced with 100:

Router B

```
RouterB# show ip bgp 1.1.1.1
BGP routing table entry for 1.1.1.1/32, version 3
Paths: (1 available, best #1, table default)
   Not advertised to any peer
   100 1001 100 100 100 1002 1003 1003
     172.30.0.6 from 172.30.0.6 (192.168.1.2)
       Origin IGP, localpref 100, valid, external, best RouterB#
```

Router B

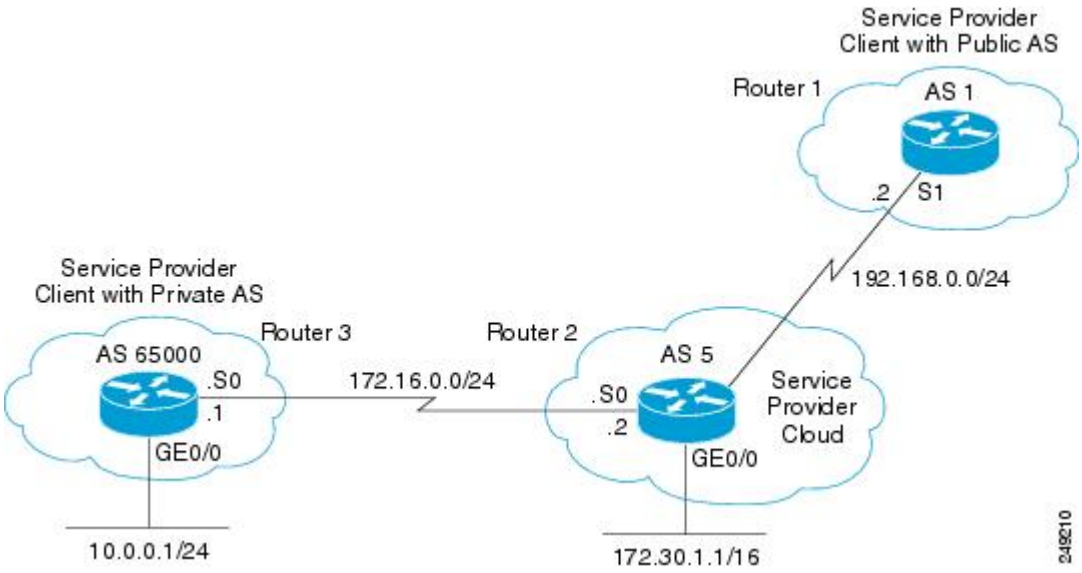
```
router bgp 200
 bgp log-neighbor-changes
 neighbor 172.30.0.6 remote-as 100
 no auto-summary
```

Example Removing Private ASNs (Cisco IOS XE Release 2)

In this example, Router 3 uses private ASN 65000. Router 1 and Router 2 use public ASNs AS 1 and AS 5 respectively.

The figure below illustrates Router 2 belonging to a service provider, with Router 1 and Router 3 as its clients.

Figure 66: Removing Private AS Numbers



In this example, Router 2, belonging to the Service Provider, removes private AS numbers as follows.

1. Router 3 advertises the network 10.0.0.0/24 with the AS path attribute 65000 to Router 2.
2. Router 2 receives the update from Router 3 and makes an entry for the network 10.0.0.0/24 in its routing table with the next hop as 172.16.0.1 (serial interface S0 on Router 3).
3. Router 2 (service provider device), when configured with the **neighbor 192.168.0.2 remove-private-as** command, strips off the private AS number and constructs a new update packet with its own AS number as the AS path attribute for the 10.0.0.0/24 network and sends the packet to Router 1.
4. Router 1 receives the eBGP update for network 10.0.0.0/24 and makes an entry in its routing table with the next hop as 192.168.0.1 (serial interface S1 on Router 2). The AS path attribute for this network as seen on Router 1 is AS 5 (Router 2). Thus, the private AS numbers are prevented from entering the BGP tables of the Internet.

The configurations of Router 3, Router 2, and Router 1 follow.

Router 3

```
interface gigabitethernet 0/0
 ip address 10.0.0.1 255.255.255.0
!
interface Serial 0
 ip address 172.16.0.1 255.255.255.0
!
router bgp 65000
 network 10.0.0.0 mask 255.255.255.0
 neighbor 172.16.0.2 remote-as 5
!---Configures Router 2 as an eBGP neighbor in public AS 5.
!
end
```

Router 2

```
interface gigabitethernet 0/0
 ip address 172.30.1.1 255.255.0.0
!
interface Serial 0
 ip address 172.16.0.2 255.255.255.0
!
interface Serial 1
 ip address 192.168.0.1 255.255.255.0
!
router bgp 5
 network 172.30.0.0
 network 192.168.0.0
 neighbor 172.16.0.1 remote-as 65000
!---Configures Router 3 as an eBGP neighbor in private AS 65000.
 neighbor 192.168.0.2 remote-as 1
!---Configures Router 1 as an eBGP neighbor in public AS 1.
 neighbor 192.168.0.2 remove-private-as
!---Removes the private AS numbers from outgoing eBGP updates.
!
end
```

Router 1

```
version 12.2
!
!
interface Serial 0
 ip address 192.168.0.2 255.255.255.0
!
router bgp 1
 neighbor 192.168.0.1 remote-as 5
!---Configures Router 2 as an eBGP neighbor in public AS 5.
!
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	*Cisco IOS IP Routing: BGP Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

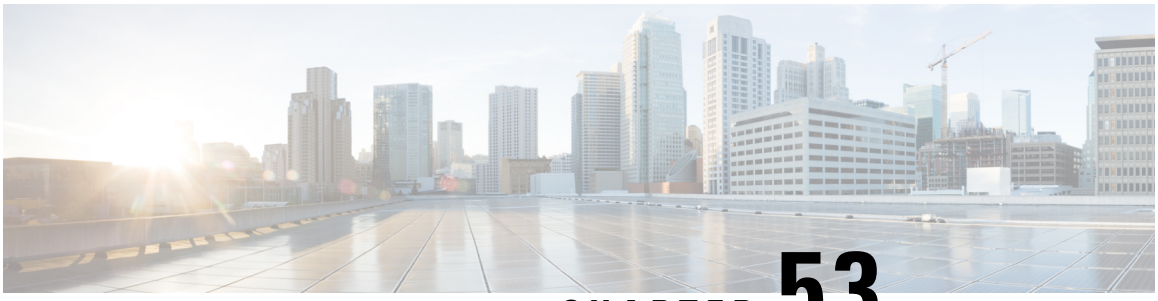
Feature Information for Removing and Replacing Private ASNs from the AS Path

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 78: Feature Information for BGP--Remove/Replace Private AS

Feature Name	Releases	Feature Information
BGP--Remove/Replace Private AS Filter	Cisco IOS XE Release 3.1S	Private autonomous system (AS) numbers are used by ISPs and customer networks to conserve globally unique AS numbers. Private AS numbers cannot be used to access the globalInternet because they are not unique. AS numbers appear in eBGP AS paths in routing tables. Removing private AS numbers from the AS path is necessary if you have been using private AS numbers and you want to access the global Internet. The following command is modified: • **neighbor remove-private-as**



CHAPTER 53

Configuring BGP Neighbor Session Options

This module describes configuration tasks to configure various options involving Border Gateway Protocol (BGP) neighbor peer sessions. BGP is an interdomain routing protocol designed to provide loop-free routing between organizations. This module contains tasks that use BGP neighbor session commands to configure:

- Options to help an autonomous system migration
- TTL Security Check, a lightweight security mechanism to protect External BGP (eBGP) peering sessions from CPU-utilization-based attacks

Information About Configuring BGP Neighbor Session Options

BGP Neighbor Sessions

BGP is mainly used to connect a local network to an external network to gain access to the Internet or to connect to other organizations. A BGP-speaking router does not discover another BGP-speaking device automatically. A network administrator usually manually configures the relationships between BGP-speaking routers.

A BGP neighbor device is a BGP-speaking router that has an active TCP connection to another BGP-speaking device. This relationship between BGP devices is often referred to as a peer instead of neighbor because a neighbor may imply the idea that the BGP devices are directly connected with no other router in between. Configuring BGP neighbor or peer sessions uses BGP neighbor session commands so this module uses the term “neighbor” over “peer.”

BGP Support for Fast Peering Session Deactivation

BGP Hold Timer

By default, the BGP hold timer is set to run every 180 seconds in Cisco software. This timer value is set as the default to protect the BGP routing process from instability that can be caused by peering sessions with other routing protocols. BGP devices typically carry large routing tables, so frequent session resets are not desirable.

BGP Fast Peering Session Deactivation

BGP fast peering session deactivation improves BGP convergence and response time to adjacency changes with BGP neighbors. This feature is event driven and configured on a per-neighbor basis. When this feature is enabled, BGP will monitor the peering session with the specified neighbor. Adjacency changes are detected and terminated peering sessions are deactivated in between the default or configured BGP scanning interval.

Selective Address Tracking for BGP Fast Session Deactivation

In Cisco IOS Release 12.4(4)T, 12.2(31)SB, 12.2(33)SRB, and later releases, the BGP Selective Address Tracking feature introduced the use of a route map with BGP fast session deactivation. The **route-map** keyword and *map-name* argument are used with the **neighbor fall-over** BGP neighbor session command to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset. The route map is not used for session establishment.

Note The **neighbor fall-over** command is not supported in Cisco IOS Release 15.0(1)SY. The **route-map** and *map-name* keyword-argument pair in the **bgp nexthop** command are not supported in Cisco IOS Release 15.0(1)SY.

Note Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

BFD Support of BGP IPv6 Neighbors

In Cisco IOS Release 15.1(2)S and later releases, Bidirectional Forwarding Detection (BFD) can be used to track fast forwarding path failure of BGP neighbors that have an IPv6 address. BFD is a detection protocol that is designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. BFD provides faster reconvergence time for BGP after a forwarding path failure.

TTL Security Check for BGP Neighbor Sessions

BGP Support for the TTL Security Check

When implemented for BGP, the TTL Security Check feature introduces a lightweight security mechanism to protect eBGP neighbor sessions from CPU utilization-based attacks. These types of attacks are typically brute force Denial of Service (DoS) attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses.

The TTL Security Check feature protects the eBGP neighbor session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each eBGP neighbor session. If the value in the TTL field of the incoming IP packet is greater than or equal to the locally configured value, the IP packet is accepted and processed normally. If the TTL value in the IP packet is less than the locally configured value, the packet is silently discarded and no Internet Control Message Protocol (ICMP) message is generated. This is designed behavior; a response to a forged packet is unnecessary.

Although it is possible to forge the TTL field in an IP packet header, accurately forging the TTL count to match the TTL count from a trusted peer is impossible unless the network to which the trusted peer belongs has been compromised.

The TTL Security Check feature supports both directly connected neighbor sessions and multihop eBGP neighbor sessions. The BGP neighbor session is not affected by incoming packets that contain invalid TTL values. The BGP neighbor session will remain open, and the router will silently discard the invalid packet. The BGP session, however, can still expire if keepalive packets are not received before the session timer expires.

TTL Security Check for BGP Neighbor Sessions

The BGP Support for TTL Security Check feature is configured with the **neighbor ttl-security** command in router configuration mode or address family configuration mode. When this feature is enabled, BGP will establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the peering session. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router. The *hop-count* argument is used to configure the maximum number of hops that separate the two peers. The TTL value is determined by the router from the configured hop count. The value for this argument is a number from 1 to 254.

TTL Security Check Support for Multihop BGP Neighbor Sessions

The BGP Support for TTL Security Check feature supports both directly connected neighbor sessions and multihop neighbor sessions. When this feature is configured for a multihop neighbor session, the **neighbor ebgp-multihop** router configuration command cannot be configured and is not needed to establish the neighbor session. These commands are mutually exclusive, and only one command is required to establish a multihop neighbor session. If you attempt to configure both commands for the same peering session, an error message will be displayed in the console.

To configure this feature for an existing multihop session, you must first disable the existing neighbor session with the **no neighbor ebgp-multihop** command. The multihop neighbor session will be restored when you enable this feature with the **neighbor ttl-security** command.

This feature should be configured on each participating router. To maximize the effectiveness of this feature, the *hop-count* argument should be strictly configured to match the number of hops between the local and external network. However, you should also consider path variation when configuring this feature for a multihop neighbor session.

Benefits of the BGP Support for TTL Security Check

The BGP Support for TTL Security Check feature provides an effective and easy-to-deploy solution to protect eBGP neighbor sessions from CPU utilization-based attacks. When this feature is enabled, a host cannot attack a BGP session if the host is not a member of the local or remote BGP network or if the host is not directly connected to a network segment between the local and remote BGP networks. This solution greatly reduces the effectiveness of DoS attacks against a BGP autonomous system.

BGP Support for TCP Path MTU Discovery per Session

Path MTU Discovery

The IP protocol family was designed to use a wide variety of transmission links. The maximum IP packet length is 65000 bytes. Most transmission links enforce a smaller maximum packet length limit, called the maximum transmission unit (MTU), which varies with the type of the transmission link. The design of IP accommodates link packet length limits by allowing intermediate routers to fragment IP packets as necessary for their outgoing links. The final destination of an IP packet is responsible for reassembling its fragments as necessary.

All TCP sessions are bounded by a limit on the number of bytes that can be transported in a single packet, and this limit is known as the maximum segment size (MSS). TCP breaks up packets into chunks in a transmit queue before passing packets down to the IP layer. A smaller MSS may not be fragmented at an IP device along the path to the destination device, but smaller packets increase the amount of bandwidth needed to transport the packets. The maximum TCP packet length is determined by both the MTU of the outbound interface on the source device and the MSS announced by the destination device during the TCP setup process.

Path MTU discovery (PMTUD) was developed as a solution to the problem of finding the optimal TCP packet length. PMTUD is an optimization (detailed in RFC 1191) wherein a TCP connection attempts to send the longest packets that will not be fragmented along the path from source to destination. It does this by using a flag, don’t fragment (DF), in the IP packet. This flag is supposed to alter the behavior of an intermediate router that cannot send the packet across a link because it is too long. Normally the flag is off, and the router should fragment the packet and send the fragments. If a router tries to forward an IP datagram, with the DF bit set, to a link that has a lower MTU than the size of the packet, the router will drop the packet and return an ICMP Destination Unreachable message to the source of this IP datagram, with the code indicating "fragmentation needed and DF set." When the source device receives the ICMP message, it will lower the send MSS, and when TCP retransmits the segment, it will use the smaller segment size.

BGP Neighbor Session TCP PMTUD

TCP path MTU discovery is enabled by default for all BGP neighbor sessions, but there are situations when you may want to disable TCP path MTU discovery for one or all BGP neighbor sessions. Although PMTUD works well for larger transmission links (for example, Packet over Sonet links), a badly configured TCP implementation or a firewall may slow or stop the TCP connections from forwarding any packets. In this type of situation, you may need to disable TCP path MTU discovery.

In Cisco software, configuration options were introduced to permit TCP path MTU discovery to be disabled, or subsequently reenabled, either for a single BGP neighbor session or for all BGP sessions. To disable the TCP path MTU discovery globally for all BGP neighbors, use the **no bgp transport path-mtu-discovery** command in router configuration mode. To disable the TCP path MTU discovery for a single neighbor, use the **no neighbor transport path-mtu-discovery** command in router configuration mode or address family configuration mode. For more details, see the “Disabling TCP Path MTU Discovery Globally for All BGP Sessions” section or the “Disabling TCP Path MTU Discovery for a Single BGP Neighbor” section.

How to Configure BGP Neighbor Session Options

Configuring Fast Session Deactivation

The tasks in this section show how to configure BGP next-hop address tracking. BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP. For more details about route dampening, see the "Configuring Internal BGP Features" module.

Configuring Fast Session Deactivation for a BGP Neighbor

Perform this task to establish a peering session with a BGP neighbor and then configure the peering session for fast session deactivation to improve the network convergence time if the peering session is deactivated.

Enabling fast session deactivation for a BGP neighbor can significantly improve BGP convergence time. However, unstable IGP peers can still introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Enter one of the following commands:
   - **address-family ipv4** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
   - **address-family ipv6** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **fall-over**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:**	Enters router configuration mode to create or configure a BGP routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 50000`	
Step 4	Enter one of the following commands: • **address-family ipv4** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] • **address-family ipv6** [**unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast vrf blue`	Enters address family configuration mode and enables IPv4 or IPv6 addressing. Perform this step when configuring fast session deactivation for a VRF address-family. **Note** Step 4 is only required if you are configuring fast session deactivation on a VRF. If you are not configuring fast session deactivation on a VRF, skip this step and perform the following commands under router BGP mode (config-router) rather than address family configuration mode (config-router-af).
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.1 remote-as 50000`	Establishes a peering session with a BGP neighbor.
Step 6	**neighbor** *ip-address* **fall-over** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 fall-over`	Configures the BGP peering to use fast session deactivation. • BGP will remove all routes learned through this peer if the session is deactivated.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Configuring Selective Address Tracking for Fast Session Deactivation

Perform this task to configure selective address tracking for fast session deactivation. The optional **route-map** keyword and *map-name* argument of the **neighbor fall-over** command are used to determine if a peering session with a BGP neighbor should be deactivated (reset) when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset.

Note Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*

4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **fall-over** [**route-map** *map-name*]
6. **exit**
7. **ip prefix-list** *list-name* [**seq** *seq-value*]{**deny** *network* / *length* | **permit** *network* / *length*} [**ge** *ge-value*] [**le** *le-value*]
8. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
9. **match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**neighbor** *ip-address* **fall-over** [**route-map** *map-name*] **Example:** `Device(config-router)# neighbor 192.168.1.2 fall-over route-map CHECK-NBR`	Applies a route map when a route to the BGP changes. • In this example, the route map named CHECK-NBR is applied when the route to neighbor 192.168.1.2 changes.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 7	**ip prefix-list** *list-name* [**seq** *seq-value*]{**deny** *network* / *length* | **permit** *network* / *length*} [**ge** *ge-value*] [**le** *le-value*]	Creates a prefix list for BGP next-hop route filtering.

	Command or Action	Purpose
	Example: `Device(config)# ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28`	• Selective next-hop route filtering supports prefix-length matching or source-protocol matching on a per-address family basis. • The example creates a prefix list named FILTER28 that permits routes only if the mask length is greater than or equal to 28.
Step 8	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map CHECK-NBR permit 10`	Configures a route map and enters route-map configuration mode. • In this example, a route map named CHECK-NBR is created. If there is an IP address match in the following **match** command, the IP address will be permitted.
Step 9	**match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*] **Example:** `Device(config-route-map)# match ip address prefix-list FILTER28`	Matches the IP addresses in the specified prefix list. • Use the *prefix-list-name* argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 10	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and enters privileged EXEC mode.

What to Do Next

After the peer policy template is created, the configuration of the peer policy template can be inherited or applied by another peer policy template. For details about peer policy inheritance, see the "Configuring Peer Policy Template Inheritance with the inherit peer-policy Command" section or the "Configuring Peer Policy Template Inheritance with the neighbor inherit peer-policy Command" section.

Configuring BFD for BGP IPv6 Neighbors

In Cisco IOS Release 15.1(2)S and later releases, Bidirectional Forwarding Detection (BFD) can be used for BGP neighbors that have an IPv6 address.

Once it has been verified that BFD neighbors are up, the **show bgp ipv6 unicast neighbors** command will indicate that BFD is being used to detect fast fallover on the specified neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ipv6 unicast-routing**
4. **ipv6 cef**
5. **interface** *type number*
6. **ipv6 address** *ipv6-address* / *prefix-length*
7. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *multiplier-value*
8. **no shutdown**
9. **exit**
10. **router bgp** *autonomous-system-number*
11. **no bgp default ipv4-unicast**
12. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
13. **neighbor** *ipv6-address* **remote-as** *autonomous-system-number*
14. **neighbor** *ipv6-address* **fall-over bfd**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**ipv6 cef** **Example:** `Device(config)# ipv6 cef`	Enables Cisco Express Forwarding for IPv6.
Step 5	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Configures an interface type and number.
Step 6	**ipv6 address** *ipv6-address* / *prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1:1::1/64`	Configures an IPv6 address and enables IPv6 processing on an interface.

	Command or Action	Purpose
Step 7	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *multiplier-value* **Example:** `Device(config-if)# bfd interval 500 min_rx 500 multiplier 3`	Sets the baseline BFD session parameters on an interface.
Step 8	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts an interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 11	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	Disables the default IPv4 unicast address family for establishing peering sessions. • We recommend configuring this command in the global scope.
Step 12	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode and enables IPv6 addressing.
Step 13	**neighbor** *ipv6-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 2001:DB8:2:1::4 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv6 BGP neighbor table of the local router.
Step 14	**neighbor** *ipv6-address* **fall-over bfd** **Example:** `Device(config-router-af)# neighbor 2001:DB8:2:1::4 fall-over bfd`	Enables BGP to monitor the peering session of an IPv6 neighbor using BFD.

	Command or Action	Purpose
Step 15	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Configuring the TTL Security Check for BGP Neighbor Sessions

Perform this task to allow BGP to establish or maintain a session only if the TTL value in the IP packet header is equal to or greater than the TTL value configured for the BGP neighbor session.

Before you begin

- To maximize the effectiveness of the BGP Support for TTL Security Check feature, we recommend that you configure it on each participating router. Enabling this feature secures the eBGP session in the incoming direction only and has no effect on outgoing IP packets or the remote router.

Note

- The **neighbor ebgp-multihop** command is not needed when the BGP Support for TTL Security Check feature is configured for a multihop neighbor session and should be disabled before configuring this feature.
- The effectiveness of the BGP Support for TTL Security Check feature is reduced in large-diameter multihop peerings. In the event of a CPU utilization-based attack against a BGP router that is configured for large-diameter peering, you may still need to shut down the affected neighbor sessions to handle the attack.
- This feature is not effective against attacks from a peer that has been compromised inside of the local and remote network. This restriction also includes peers that are on the network segment between the local and remote network.

SUMMARY STEPS

1. **enable**
2. **trace** [*protocol*] *destination*
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **neighbor** *ip-address* **ttl-security hops** *hop-count*
6. **end**
7. **show running-config**
8. **show ip bgp neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**trace** [*protocol*] *destination* **Example:** `Device# trace ip 10.1.1.1`	Discovers the routes of the specified protocol that packets will actually take when traveling to their destination. • Enter the **trace** command to determine the number of hops to the specified peer.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode, and creates a BGP routing process.
Step 5	**neighbor** *ip-address* **ttl-security hops** *hop-count* **Example:** `Device(config-router)# neighbor 10.1.1.1 ttl-security hops 2`	Configures the maximum number of hops that separate two peers. • The *hop-count* argument is set to the number of hops that separate the local and remote peer. If the expected TTL value in the IP packet header is 254, then the number 1 should be configured for the *hop-count* argument. The range of values is a number from 1 to 254. • When the BGP Support for TTL Security Check feature is enabled, BGP will accept incoming IP packets with a TTL value that is equal to or greater than the expected TTL value. Packets that are not accepted are discarded. • The example configuration sets the expected incoming TTL value to at least 253, which is 255 minus the TTL value of 2, and this is the minimum TTL value expected from the BGP peer. The local router will accept the peering session from the 10.1.1.1 neighbor only if it is one or two hops away.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 7	**show running-config** **Example:**	(Optional) Displays the contents of the currently running configuration file.

	Command or Action	Purpose
	`Device# show running-config | begin bgp`	• The output of this command displays the configuration of the **neighbor ttl-security** command for each peer under the BGP configuration section of output. That section includes the neighbor address and the configured hop count. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 8	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# show ip bgp neighbors 10.4.9.5`	(Optional) Displays information about the TCP and BGP connections to neighbors. • This command displays "External BGP neighbor may be up to *number* hops away" when the BGP Support for TTL Security Check feature is enabled. The *number* value represents the hop count. It is a number from 1 to 254. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The configuration of the BGP Support for TTL Security Check feature can be verified with the **show running-config** and **show ip bgp neighbors** commands. This feature is configured locally on each peer, so there is no remote configuration to verify.

The following is sample output from the **show running-config** command. The output shows that neighbor 10.1.1.1 is configured to establish or maintain the neighbor session only if the expected TTL count in the incoming IP packet is 253 or 254.

```
Router# show running-config
| begin bgp

router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.1.1.1 remote-as 55000
 neighbor 10.1.1.1 ttl-security hops 2
 no auto-summary
.
.
.
```

The following is sample output from the **show ip bgp neighbors** command. The output shows that the local router will accept packets from the 10.1.1.1 neighbor if it is no more than 2 hops away. The configuration of this feature is displayed in the address family section of the output. The relevant line is shown in bold in the output.

```
Router# show ip bgp neighbors 10.1.1.1
BGP neighbor is 10.1.1.1,  remote AS 55000, external link
  BGP version 4, remote router ID 10.2.2.22
  BGP state = Established, up for 00:59:21
  Last read 00:00:21, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  2          2
    Notifications:          0          0
    Updates:                0          0
    Keepalives:           226        227
    Route Refresh:          0          0
    Total:                228        229
  Default minimum time between advertisement runs is 5 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue sizes : 0 self, 0 replicated
  Index 1, Offset 0, Mask 0x2
  Member of update-group 1
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:             0          0
    Prefixes Total:               0          0
    Implicit Withdraw:            0          0
    Explicit Withdraw:            0          0
    Used as bestpath:           n/a          0
    Used as multipath:          n/a          0
                                 Outbound    Inbound
  Local Policy Denied Prefixes:  --------    -------
    Total:                              0          0
  Number of NLRIs in the update sent: max 0, min 0
  Connections established 2; dropped 1
  Last reset 00:59:50, due to User reset
  External BGP neighbor may be up to 2 hops away.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.2.2.22, Local port: 179
Foreign host: 10.1.1.1, Foreign port: 11001
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0xCC28EC):
Timer          Starts    Wakeups            Next
Retrans            63          0             0x0
TimeWait            0          0             0x0
AckHold            62         50             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss:  712702676  snduna:  712703881  sndnxt:  712703881     sndwnd:  15180
irs: 2255946817  rcvnxt: 2255948041  rcvwnd:      15161  delrcvwnd:   1223
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs

Datagrams (max data segment is 1460 bytes):
Rcvd: 76 (out of order: 0), with data: 63, total data bytes: 1223
Sent: 113 (retransmit: 0, fastretransmit: 0), with data: 62, total data bytes: 4
```

Configuring BGP Support for TCP Path MTU Discovery per Session

This section contains the following tasks:

Disabling TCP Path MTU Discovery Globally for All BGP Sessions

Perform this task to disable TCP path MTU discovery for all BGP sessions. TCP path MTU discovery is enabled by default when you configure BGP sessions, but we recommend that you enter the **show ip bgp neighbors** command to ensure that TCP path MTU discovery is enabled.

Before you begin

This task assumes that you have previously configured BGP neighbors with active TCP connections.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** [*ip-address*]
3. **configure terminal**
4. **router bgp** *autonomous-system-number*
5. **no bgp transport path-mtu-discovery**
6. **end**
7. **show ip bgp neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • Use this command to determine whether BGP neighbors have TCP path MTU discovery enabled. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**router bgp** *autonomous-system-number* **Example:**	Enters router configuration mode to create or configure a BGP routing process.

	Command or Action	Purpose
	`Device(config)# router bgp 50000`	
Step 5	**no bgp transport path-mtu-discovery** **Example:** `Device(config-router)# no bgp transport path-mtu-discovery`	Disables TCP path MTU discovery for all BGP sessions.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 7	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will not display that any neighbors have TCP path MTU enabled. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output from the **show ip bgp neighbors** command shows that TCP path MTU discovery is enabled for BGP neighbors. Two entries in the output—Transport(tcp) path-mtu-discovery is enabled and path mtu capable—show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors

BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 3; dropped 2
  Last reset 00:00:35, due to Router ID changed
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
```

```
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

The following is sample output from the **show ip bgp neighbors** command after the **no bgp transport path-mtu-discovery** command has been entered. Note that the path mtu entries are missing.

```
Router# show ip bgp neighbors

BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 3; dropped 2
  Last reset 00:00:35, due to Router ID changed
.
.
.
SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle
```

Disabling TCP Path MTU Discovery for a Single BGP Neighbor

Perform this task to establish a peering session with an internal BGP (iBGP) neighbor and then disable TCP path MTU discovery for the BGP neighbor session. The **neighbor transport** command can be used in router configuration mode or address family configuration mode.

Before you begin

This task assumes that you know that TCP path MTU discovery is enabled by default for all your BGP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
5. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address*| *peer-group-name*} **activate**
7. **no neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**}
8. **end**
9. **show ip bgp neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. • The example creates an IPv4 unicast address family session.
Step 5	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 6	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 172.16.1.1 activate`	Activates the neighbor under the IPv4 address family.
Step 7	**no neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**} **Example:** `Device(config-router-af)# no neighbor 172.16.1.1 transport path-mtu-discovery`	Disables TCP path MTU discovery for a single BGP neighbor. • In this example, TCP path MTU discovery is disabled for the neighbor at 172.16.1.1.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show ip bgp neighbors** **Example:** `Device# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will not display that the neighbor has TCP path MTU discovery enabled. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output shows that TCP path MTU discovery has been disabled for BGP neighbor 172.16.1.1 but that it is still enabled for BGP neighbor 192.168.2.2. Two entries in the output—Transport(tcp) path-mtu-discovery is enabled and path mtu capable—show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
BGP neighbor is 172.16.1.1,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.17.1.99
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.1
  Address tracking requires at least a /24 route to the peer
  Connections established 1; dropped 0
  Last reset never
.
.
.
SRTT: 165 ms, RTTO: 1172 ms, RTV: 1007 ms, KRTT: 0 ms
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle
.
.
.
BGP neighbor is 192.168.2.2,  remote AS 50000, external link
  BGP version 4, remote router ID 10.2.2.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 192.168.2.2
  Address tracking requires at least a /24 route to the peer
  Connections established 2; dropped 1
  Last reset 00:05:11, due to User reset
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 210 ms, RTTO: 904 ms, RTV: 694 ms, KRTT: 0 ms
```

```
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

Enabling TCP Path MTU Discovery Globally for All BGP Sessions

Perform this task to enable TCP path MTU discovery for all BGP sessions. TCP path MTU discovery is enabled by default when you configure BGP sessions, but if the BGP Support for TCP Path MTU Discovery per Session feature has been disabled, you can use this task to reenable it. To verify that TCP path MTU discovery is enabled, use the **show ip bgp neighbors** command.

Before you begin

This task assumes that you have previously configured BGP neighbors with active TCP connections.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp transport path-mtu-discovery**
5. **end**
6. **show ip bgp neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**bgp transport path-mtu-discovery** **Example:** `Device(config-router)# bgp transport path-mtu-discovery`	Enables TCP path MTU discovery for all BGP sessions.
Step 5	**end** **Example:**	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router)# end`	
Step 6	**show ip bgp neighbors** **Example:** `Device# show ip bgp neighbors`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will show that all neighbors have TCP path MTU discovery enabled. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output from the **show ip bgp neighbors** command shows that TCP path MTU discovery is enabled for BGP neighbors. Two entries in the output—Transport(tcp) path-mtu-discovery is enabled and path mtu capable—show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors
BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.16.1.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 3; dropped 2
  Last reset 00:00:35, due to Router ID changed
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 146 ms, RTTO: 1283 ms, RTV: 1137 ms, KRTT: 0 ms
minRTT: 8 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

Enabling TCP Path MTU Discovery for a Single BGP Neighbor

Perform this task to establish a peering session with an eBGP neighbor and then enable TCP path MTU discovery for the BGP neighbor session. The **neighbor transport** command can be used in router configuration mode or address family configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **router bgp** *autonomous-system-number*
4. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
5. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address*| *peer-group-name*} **activate**
7. **neighbor** {*ip-address*| *peer-group-name*} **transport**{**connection-mode** | **path-mtu-discovery**}
8. **end**
9. **show ip bgp neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family** {**ipv4** [**mdt**	**multicast**	**unicast** [**vrf** *vrf-name*]	**vrf** *vrf-name*]	**vpnv4** [**unicast**]} **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. • The example creates an IPv4 unicast address family session.
Step 5	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.2.2 remote-as 50000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 6	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.2 activate`	Activates the neighbor under the IPv4 address family.
Step 7	**neighbor** {*ip-address*	*peer-group-name*} **transport**{**connection-mode**	**path-mtu-discovery**} **Example:**	Enables TCP path MTU discovery for a single BGP neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.2.2 transport path-mtu-discovery`	
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 9	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# show ip bgp neighbors 192.168.2.2`	(Optional) Displays information about the TCP and BGP connections to neighbors. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following sample output from the **show ip bgp neighbors** command shows that TCP path MTU discovery is enabled for the BGP neighbor at 192.168.2.2. Two entries in the output—Transport(tcp) path-mtu-discovery is enabled and path-mtu capable—show that TCP path MTU discovery is enabled.

```
Router# show ip bgp neighbors 192.168.2.2
BGP neighbor is 192.168.2.2,  remote AS 50000, external link
  BGP version 4, remote router ID 10.2.2.99
.
.
.
 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
.
.
.
  Address tracking is enabled, the RIB does have a route to 192.168.2.2
  Address tracking requires at least a /24 route to the peer
  Connections established 2; dropped 1
  Last reset 00:05:11, due to User reset
  Transport(tcp) path-mtu-discovery is enabled
.
.
.
SRTT: 210 ms, RTTO: 904 ms, RTV: 694 ms, KRTT: 0 ms
minRTT: 20 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, retransmission timeout, nagle, path mtu capable
```

Configuration Examples for BGP Neighbor Session Options

Example: Configuring Fast Session Deactivation for a BGP Neighbor

In the following example, the BGP routing process is configured on device A and device B to monitor and use fast peering session deactivation for the neighbor session between the two devices. Although fast peering session deactivation is not required at both devices in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

Device A

```
router bgp 40000
 neighbor 192.168.1.1 remote-as 45000
 neighbor 192.168.1.1 fall-over
 end
```

Device B

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 fall-over
 end
```

Example: Configuring Selective Address Tracking for Fast Session Deactivation

The following example shows how to configure the BGP peering session to be reset if a route with a prefix of /28 or a more specific route to a peer destination is no longer available:

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 fall-over route-map CHECK-NBR
 exit
ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28
route-map CHECK-NBR permit 10
 match ip address prefix-list FILTER28
 end
```

Example: Configuring BFD for a BGP IPv6 Neighbor

The following example configures FastEthernet interface 0/1 with the IPv6 address 2001:DB8:4:1::1. Bidirectional Forwarding Detection (BFD) is configured for the BGP neighbor at 2001:DB8:5:1::2. BFD will track forwarding path failure of the BGP neighbor and provide faster reconvergence time for BGP after a forwarding path failure.

```
ipv6 unicast-routing
ipv6 cef
interface fastethernet 0/1
 ipv6 address 2001:DB8:4:1::1/64
 bfd interval 500 min_rx 500 multiplier 3
 no shutdown
```

```
 exit
router bgp 65000
 no bgp default ipv4-unicast
 address-family ipv6 unicast
 neighbor 2001:DB8:5:1::2 remote-as 65001
 neighbor 2001:DB8:5:1::2 fall-over bfd
 end
```

Example: Configuring the TTL-Security Check

The example configurations in this section show how to configure the BGP Support for TTL Security Check feature.

The following example uses the **trace** command to determine the hop count to an eBGP peer. The hop count number is displayed in the output for each networking device that IP packets traverse to reach the specified neighbor. In the following example, the hop count for the 10.1.1.1 neighbor is 1.

```
Router# trace ip 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.1.1.1
  1 10.1.1.1 0 msec *  0 msec
```

The following example sets the hop count to 2 for the 10.1.1.1 neighbor. Because the hop-count argument is set to 2, BGP will accept only IP packets with a TTL count in the header that is equal to or greater than 253.

```
Router(config-router)# neighbor 10.1.1.1 ttl-security hops 2
```

Examples: Configuring BGP Support for TCP Path MTU Discovery per Session

This section contains the following configuration examples:

Example: Disabling TCP Path MTU Discovery Globally for All BGP Sessions

The following example shows how to disable TCP path MTU discovery for all BGP neighbor sessions. Use the **show ip bgp neighbors** command to verify that TCP path MTU discovery has been disabled.

```
enable
configure terminal
 router bgp 45000
  no bgp transport path-mtu-discovery
  end
show ip bgp neighbors
```

Example: Disabling TCP Path MTU Discovery for a Single BGP Neighbor

The following example shows how to disable TCP path MTU discovery for an eBGP neighbor at 192.168.2.2:

```
enable
configure terminal
 router bgp 45000
  neighbor 192.168.2.2 remote-as 50000
  neighbor 192.168.2.2 activate
  no neighbor 192.168.2.2 transport path-mtu-discovery
  end
show ip bgp neighbors 192.168.2.2
```

Example: Enabling TCP Path MTU Discovery Globally for All BGP Sessions

The following example shows how to enable TCP path MTU discovery for all BGP neighbor sessions. Use the **show ip bgp neighbors** command to verify that TCP path MTU discovery has been enabled.

```
enable
configure terminal
 router bgp 45000
  bgp transport path-mtu-discovery
  end
show ip bgp neighbors
```

Example: Enabling TCP Path MTU Discovery for a Single BGP Neighbor

The following example shows how to enable TCP path MTU discovery for an eBGP neighbor at 192.168.2.2. Use the **show ip bgp neighbors** command to verify that TCP path MTU discovery has been enabled.

```
enable
configure terminal
 router bgp 45000
  neighbor 192.168.2.2 remote-as 50000
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 transport path-mtu-discovery
  end
show ip bgp neighbors 192.168.2.2
```

Where to Go Next

For information about advertising the bandwidth of an autonomous system exit link as an extended community, refer to the “BGP Link Bandwidth” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
Overview of Cisco BGP conceptual information with links to all the individual BGP modules	“Cisco BGP Overview” module
Conceptual and configuration details for basic BGP tasks	“Configuring a Basic BGP Network” module
Conceptual and configuration details for advanced BGP tasks	“Configuring Advanced BGP Features” module
Bidirectional Forwarding Detection configuration tasks	*IP Routing: BFD Configuration Guide*

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1191	*Path MTU Discovery*
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

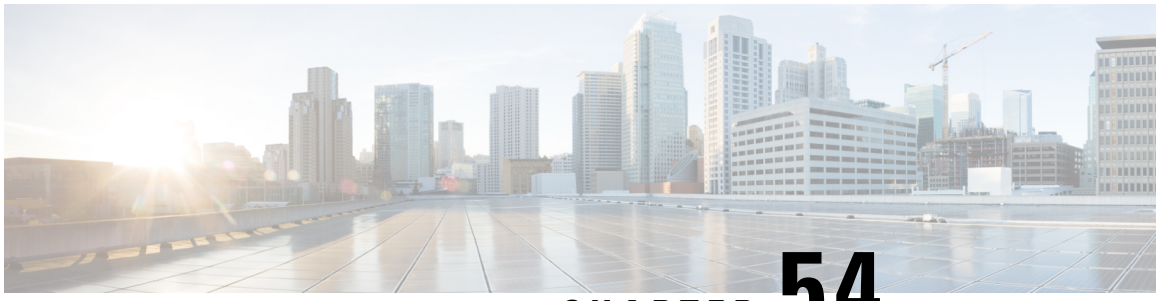
Feature Information for Configuring BGP Neighbor Session Options

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 79: Feature Information for Configuring BGP Neighbor Session Options Features

Feature Name	Releases	Feature Information
BGP Support for TCP Path MTU Discovery per Session	12.2(33)SRA 12.2(31)SB 12.2(33)SXH 12.4(20)T 15.0(1)S	BGP support for TCP path maximum transmission unit (MTU) discovery introduced the ability for BGP to automatically discover the best TCP path MTU for each BGP session. The TCP path MTU is enabled by default for all BGP neighbor sessions, but you can disable, and subsequently enable, the TCP path MTU globally for all BGP sessions or for an individual BGP neighbor session. The following commands were introduced or modified by this feature: **bgp transport**, **neighbor transport**, **show ip bgp neighbors**.
BGP Support for TTL Security Check	12.0(27)S 12.3(7)T 12.2(25)S 12.2(18)SXE 15.0(1)S	The BGP Support for TTL Security Check feature introduced a lightweight security mechanism to protect external Border Gateway Protocol (eBGP) peering sessions from CPU utilization-based attacks using forged IP packets. Enabling this feature prevents attempts to hijack the eBGP peering session by a host on a network segment that is not part of either BGP network or by a host on a network segment that is not between the eBGP peers. The following commands were introduced or modified by this feature: **neighbor ttl-security**, **show ip bgp neighbors**.
BGP IPv6 Client for Single-Hop BFD	15.1(2)S 15.2(3)T 15.2(4)S	Bidirectional Forwarding Detection (BFD) can be used to track fast forwarding path failure of BGP neighbors that use an IPv6 address. The following command was modified by this feature: **neighbor fall-over**. In Cisco IOS Release 15.2(4)S, support was added for the Cisco 7200 series router.



CHAPTER 54

BGP Neighbor Policy

The BGP Neighbor Policy feature introduces new keywords to two existing commands to display information about local and inherited policies. When BGP neighbors use multiple levels of peer templates, it can be difficult to determine which policies are applied to the neighbor. Inherited policies are policies that the neighbor inherits from a peer group or a peer policy template.

Information About BGP Neighbor Policy

Benefit of BGP Neighbor Policy Feature

The BGP Neighbor Policy feature introduces new keywords to the **show ip bgp neighbors policy** command and the **show ip bgp template peer-policy** command to display information about local and inherited policies. When BGP neighbors use multiple levels of peer templates, it can be difficult to determine which policies are applied to the neighbor. Inherited policies are policies that the neighbor inherits from a peer group or a peer policy template.

How to Display BGP Neighbor Policy Information

Displaying BGP Neighbor Policy Information

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** { *ip-address* | *ipv6-address* } **policy** [**detail**]
3. **show ip bgp template peer-policy** [*policy-template-name* [**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp neighbors** { *ip-address* | *ipv6-address* } **policy** [**detail**] **Example:** `Device# show ip bgp neighbors 192.168.2.3 policy detail`	Displays the policies applied to the specified neighbor.
Step 3	**show ip bgp template peer-policy** [*policy-template-name* [**detail**]] **Example:** `Device# show ip bgp template peer-policy`	Displays the locally configured peer policy templates.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

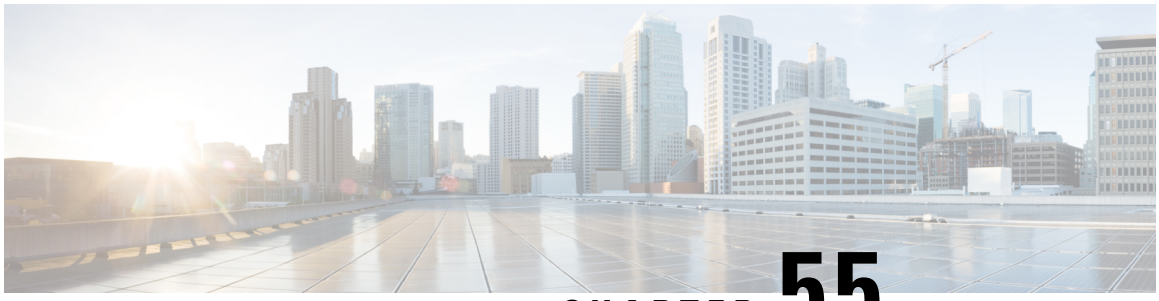
Feature Information for BGP Neighbor Policy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 80: Feature Information for BGP Neighbor Policy

Feature Name	Releases	Feature Information
BGP Neighbor Policy		The BGP Neighbor Policy feature introduces new keywords to two existing commands to display information about local and inherited policies. When BGP neighbors use multiple levels of peer templates, it can be difficult to determine which policies are applied to the neighbor. Inherited policies are policies that the neighbor inherits from a peer-group or a peer-policy template. The following commands were modified: **show ip bgp neighbors**, and **show ip bgp template peer-policy**.



CHAPTER 55

BGP Dynamic Neighbors

Border Gateway Protocol (BGP) dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups.

Information About BGP Dynamic Neighbors

Overview

Support for the BGP Dynamic Neighbors feature was introduced in Cisco IOS Release 12.2(33)SXH on the Cisco Catalyst 6500 series switches. BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups.

In Cisco IOS XE Denali 16.3 release, support for BGP dynamic neighbors was extended to IPv6 BGP peering with VRF support.

From Cisco IOS XE Dublin 17.11.1a release, support for BGP dynamic neighbors is extended to the following address families:

- Layer 2 VPN Ethernet VPN (EVPN)
- Layer 2 VPN Virtual Private LAN Service (VPLS)
- IPv4 FlowSpec
- IPv4 MDT
- IPv4 Multicast
- IPv4 Multicast VPN (MVPN)
- IPv6 FlowSpec
- IPv6 Multicast
- IPv6 Multicast VPN (MVPN)

- Link-State
- Network Service Access Point (NSAP)
- RT-filter

After a subnet range is configured for a BGP peer group and a TCP session is initiated by another router for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. After the initial configuration of subnet ranges and activation of the peer group (referred to as a *listen range group*), dynamic BGP neighbor creation does not require any further CLI configuration on the initial router. Other routers can establish a BGP session with the listening router, but the initial router need not establish a BGP session to other routers if the IP address of the remote peer used for the BGP session is not within the configured range.

To support the BGP Dynamic Neighbors feature, the output for the **show ip bgp neighbors**, **show ip bgp peer-group**, and **show ip bgp summary** commands was updated to display information about dynamic neighbors.

A dynamic BGP neighbor will inherit any configuration for the peer group. In larger BGP networks, implementing BGP dynamic neighbors can reduce the amount and complexity of CLI configuration and save CPU and memory usage.

Block BGP Dynamic Neighbor Sessions

From Cisco IOS XE Amsterdam 17.2.1, you can block a router from establishing BGP dynamic neighbor sessions with certain nodes in a BGP peer group. Identify a target nodes using its IP address. To block a router from establishing a BGP dynamic neighbor session to a node, use the **bgp listen block** {*ipv4-address*|*ipv6-address*} command.

Note Use the **bgp listen block** {*ipv4-address* | *ipv6-address*} command in router BGP mode to exclude a neighbour if you require a static peer in the listen range. This permits the listen subnet range to contain both static and dynamic peers.

When you block a router from establishing a BGP dynamic neighbor session to a node, any existing BGP dynamic neighbor session between the router and the node is terminated, and the router does not make future attempts to establish a BGP dynamic neighbor session with the node. The block command does not impact static BGP neighbor sessions.

Related Topics

Block BGP Dynamic Neighbor Session Establishment with a Node, on page 918
View Blocked BGP Dynamic Neighbor Sessions, on page 918
Debug Blocked BGP Dynamic Neighbor Sessions, on page 919

How to Configure BGP Dynamic Neighbors

Implementing BGP Dynamic Neighbors Using Subnet Ranges

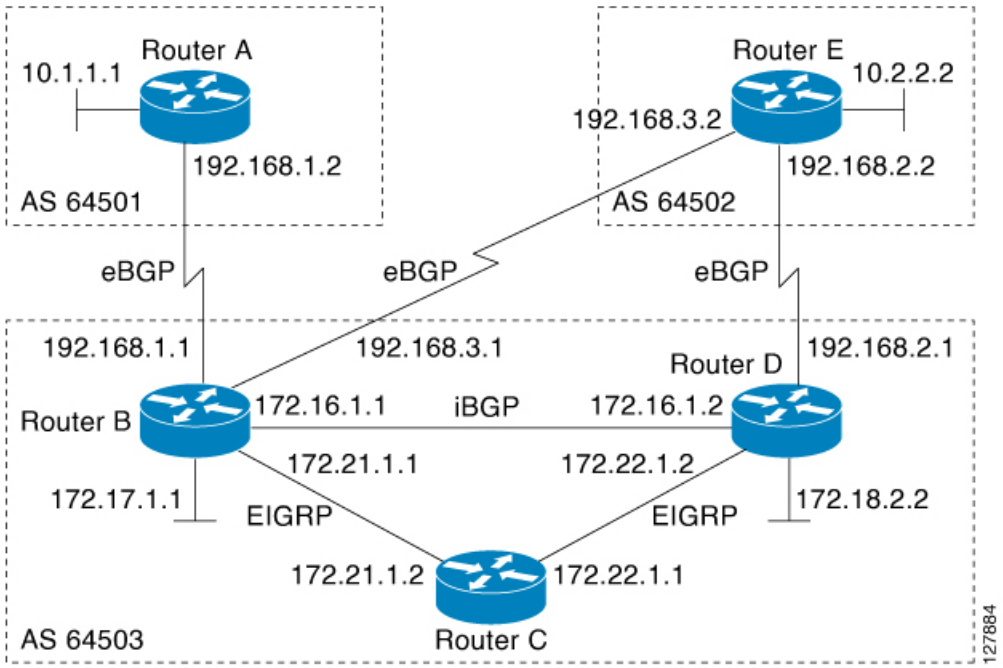
In Cisco IOS Release 12.2(33)SXH, support for BGP dynamic neighbors was introduced. Perform this task to implement the dynamic creation of BGP neighbors using subnet ranges.

In this task, a BGP peer group is created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer group is added to the BGP neighbor table of the local router, and an alternate autonomous system number is also configured. The peer group is activated under the IPv4 address family.

The next step is to move to another router—Router E in the figure below—where a BGP session is started and the neighbor router, Router B, is configured as a remote BGP peer. The peering configuration opens a TCP session and triggers Router B to create a dynamic BGP neighbor because the IP address that starts the TCP session (192.168.3.2) is within the configured subnet range for dynamic BGP peers. The task moves back to the first router, Router B, to run three **show** commands that have been modified to display dynamic BGP peer information.

Note We recommend that you keep the listen limit and listen range the same as the planned neighbor count in order to prevent unexpected peers.

Figure 67: BGP Dynamic Neighbor Topology



Before you begin

This task requires Cisco IOS Release 12.2(33)SXH, or a later release, to be running.

Note This task supports only IPv4 BGP peering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp log-neighbor-changes**
5. **neighbor** *peer-group-name* **peer-group**
6. **bgp listen** [**limit** *max-number*]
7. **bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop [***ttl*]
9. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
10. **address-family ipv4**
11. **neighbor** *peer-group-name* **activate**
12. **end**
13. Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.
14. **enable**
15. **configure terminal**
16. **router bgp** *autonomous-system-number*
17. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
18. Return to the first router.
19. **show ip bgp ipv4 summary**
20. **show ip bgp ipv4 peer-group** [*peer-group-name*] [**summary**]
21. **show ip bgp ipv4 neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `RouterB> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • The configuration is entered on router B.
Step 2	**configure terminal** **Example:** `RouterB# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `RouterB(config)# router bgp 64503`	Enters router configuration mode for the specified routing process.
Step 4	**bgp log-neighbor-changes** **Example:**	(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets.

	Command or Action	Purpose
	`RouterB(config-router)# bgp log-neighbor-changes`	• Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `RouterB(config-router)# neighbor group192 peer-group`	Creates a BGP peer group. • In this example, a peer group named group192 is created. This group will be used as a listen range group.
Step 6	**bgp listen** [**limit** *max-number*] **Example:** `RouterB(config-router)# bgp listen limit 200`	Sets a global limit of BGP dynamic subnet range neighbors. • Use the optional **limit** keyword and *max-number* argument to define the maximum number of BGP dynamic subnet range neighbors that can be created. **Note** Only the syntax applicable to this task is used in this example. For the complete syntax, see Step 7.
Step 7	**bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*] **Example:** `RouterB(config-router)# bgp listen range 192.168.0.0/16 peer-group group192`	Associates a subnet range with a BGP peer group and activates the BGP dynamic neighbors feature. • Use the optional **limit** keyword and *max-number* argument to define the maximum number of BGP dynamic neighbors that can be created. • Use the optional **range** keyword and *network* / *length* argument to define a prefix range to be associated with the specified peer group. • In this example, the prefix range 192.168.0.0/16 is associated with the listen range group named group192.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*] **Example:** `RouterB(config-router)# neighbor group192 ebgp-multihop 255`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 9	**neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*] **Example:**	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • Use the optional **alternate-as** keyword and *autonomous-system-number* argument to identify up

	Command or Action	Purpose
	`RouterB(config-router)# neighbor group192 remote-as 64501 alternate-as 64502`	to five alternate autonomous system numbers for listen range neighbors. • In this example, the peer group named group192 is configured with two possible autonomous system numbers. **Note** The **alternate-as** keyword is used only with the listen range peer groups, not with individual BGP neighbors.
Step 10	**address-family ipv4** **Example:** `RouterB(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 11	**neighbor** *peer-group-name* **activate** **Example:** `RouterB(config-router-af)# neighbor group192 activate`	Activates the neighbor or listen range peer group for the configured address family. • In this example, the neighbor 172.16.1.1 is activated for the IPv4 address family. **Note** Usually BGP peer groups cannot be activated using this command, but the listen range peer groups are a special case.
Step 12	**end** **Example:** `RouterB(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 13	Move to another router that has an interface within the subnet range for the BGP peer group configured in this task.	—
Step 14	**enable** **Example:** `RouterE> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • The configuration is entered on Router E.
Step 15	**configure terminal** **Example:** `RouterE# configure terminal`	Enters global configuration mode.
Step 16	**router bgp** *autonomous-system-number* **Example:**	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
	`RouterE(config)# router bgp 64502`	
Step 17	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*] **Example:** `RouterE(config-router)# neighbor 192.168.3.1 remote-as 64503`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • In this example, the interface (192.168.3.2 in the figure above) at Router E is with the subnet range set for the BGP listen range group, group192. When TCP opens a session to peer to Router B, Router B creates this peer dynamically.
Step 18	Return to the first router.	—
Step 19	**show ip bgp ipv4 summary** **Example:** `RouterB# show ip bgp ipv4 summary`	(Optional) Displays the BGP path, prefix, and attribute information for all connections to BGP neighbors. • In this step, the configuration has returned to Router B.
Step 20	**show ip bgp ipv4 peer-group** [*peer-group-name*] [**summary**] **Example:** `RouterB# show ip bgp ipv4 peer-group group192`	(Optional) Displays information about BGP peer groups.
Step 21	**show ip bgp ipv4 neighbors** [*ip-address*] **Example:** `RouterB# show ip bgp ipv4 neighbors 192.168.3.2`	(Optional) Displays information about BGP and TCP connections to neighbors. • In this example, information is displayed about the dynamically created neighbor at 192.168.3.2. The IP address of this BGP neighbor can be found in the output of either the **show ip bgp summary** or the **show ip bgp peer-group** command. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Examples

The following output examples were taken from Router B in the figure above after the appropriate configuration steps in this task were completed on both Router B and Router E.

The following output from the **show ip bgp summary** command shows that the BGP neighbor 192.168.3.2 was dynamically created and is a member of the listen range group, group192. The output also shows that the IP prefix range of 192.168.0.0/16 is defined for the listen range named group192.

```
Router# show ip bgp summary
BGP router identifier 192.168.3.1, local AS number 64503
BGP table version is 1, main routing table version 1
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
*192.168.3.2    4 64502       2       2        0    0    0 00:00:37        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 1/(200 max), Subnet ranges: 1
BGP peergroup group192 listen range group members:
  192.168.0.0/16
```

The following output from the **show ip bgp peer-group** command shows information about the listen range group, group192 that was configured in this task:

```
Router# show ip bgp peer-group group192
BGP peer-group is group192,  remote AS 64501
  BGP peergroup group192 listen range group members:
  192.168.0.0/16
  BGP version 4
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP neighbor is group192, peer-group external, members:
  *192.168.3.2
  Index 0, Offset 0, Mask 0x0
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
```

The following sample output from the **show ip bgp neighbors** command shows that the neighbor 192.168.3.2 is a member of the peer group, group192, and belongs to the subnet range group 192.168.0.0/16, which shows that this peer was dynamically created:

```
Router# show ip bgp neighbors 192.168.3.2
BGP neighbor is *192.168.3.2,  remote AS 64502, external link
 Member of peer-group group192 for session parameters
 Belongs to the subnet range group: 192.168.0.0/16
  BGP version 4, remote router ID 192.168.3.2
  BGP state = Established, up for 00:06:35
  Last read 00:00:33, last write 00:00:25, hold time is 180, keepalive intervals
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                0          0
    Keepalives:             7          7
    Route Refresh:          0          0
    Total:                  8          8
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
  group192 peer-group member
.
.
.
```

Configuring BGP Dynamic Neighbor Support for L2VPN EVPN

To configure BGP Dynamic Neighbor Support for L2VPN EVPN, perform these steps.

Before you begin

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp log-neighbor-changes**
5. **neighbor** *peer-group-name* **peer-group**
6. **bgp listen** [**limit** *max-number*]
7. **bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*]
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-securityhop [***ttl*]
9. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
10. **address-family l2vpn evpn**
11. **neighbor** *peer-group-name* **activate**
12. **end**
13. **enable**
14. **configure terminal**
15. **router bgp** *autonomous-system-number*
16. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
17. **address-family l2vpn evpn**
18. **neighbor** {*ip-address* | *peer-group-name*} **activate**
19. **end**
20. Return to the first router.
21. **show ip bgp l2vpn evpn summary**
22. **show ip bgp l2vpn evpn peer-group** [*peer-group-name*] [**summary**]
23. **show ip bgp l2vpn evpn neighbors** [*ip-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `RouterB> enable`	Enables privileged EXEC mode. Enter your password, if prompted. **Note** This configuration is entered on router B.
Step 2	**configure terminal** **Example:** `RouterB# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `RouterB(config)# router bgp 64501`	Enters router configuration mode for the specified routing process.
Step 4	**bgp log-neighbor-changes** **Example:** `RouterB(config-router)# bgp log-neighbor-changes`	(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets. • Use the **bgp log-neighbor-changes** command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `RouterB(config-router)# neighbor group192 peer-group`	Creates a BGP peer group. • In this example, a peer group named group192 is created. This group is used as a listen range group.
Step 6	**bgp listen** [**limit** *max-number*] **Example:** `RouterB(config-router)# bgp listen limit 200`	Sets a global limit of BGP dynamic subnet range neighbors. • Use the optional **limit** keyword and *max-number* argument to define the maximum number of BGP dynamic subnet range neighbors that can be created. **Note** Only the syntax applicable to this task is used in this example. For the complete syntax, see Step 7.
Step 7	**bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*] **Example:** `RouterB(config-router)# bgp listen range 192.168.0.0/16 peer-group group192`	Associates a subnet range with a BGP peer group and activates the BGP dynamic neighbors feature. • Use the optional **limit** keyword and *max-number* argument to define the maximum number of BGP dynamic neighbors that can be created. • Use the optional **range** keyword and *network* / *length* argument to define a prefix range to be associated with the specified peer group. • In this example, the prefix range 192.168.0.0/16 is associated with the listen range group named group192.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-securityhop [***ttl*] **Example:**	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.

	Command or Action	Purpose
	`RouterB(config-router)# neighbor group192 ttl-security hops 2`	
Step 9	**neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*] **Example:** `RouterB(config-router)# neighbor group192 remote-as 64501 alternate-as 64502`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • Use the optional **alternate-as** keyword and *autonomous-system-number* argument to identify up to five alternate autonomous system numbers for listen range neighbors. • In this example, the peer group named group192 is configured with two possible autonomous system numbers. **Note** The **alternate-as** keyword is used only with the listen range peer groups, not with individual BGP neighbors.
Step 10	**address-family l2vpn evpn** **Example:** `RouterB(config-router)# address-family l2vpn evpn`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 11	**neighbor** *peer-group-name* **activate** **Example:** `RouterB(config-router-af)# neighbor group192 activate`	Activates the neighbor or listen range peer group for the configured address family. • In this example, the neighbor 192.168.5.7 is activated for the L2VPN EVPN address family. **Note** Usually, BGP peer groups cannot be activated using **neighbor** *peer-group-name* **activate** command, but the listen range peer groups are a special case.
Step 12	**end** **Example:** `RouterB(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode. Move to another router that has an interface within the subnet range for the BGP peer group configured in this task. —
Step 13	**enable** **Example:** `RouterE> enable`	Enables privileged EXEC mode.Enter your password, if prompted. **Note** The configuration is entered on Router E.

	Command or Action	Purpose
Step 14	**configure terminal** **Example:** `RouterE# configure terminal`	Enters global configuration mode.
Step 15	**router bgp** *autonomous-system-number* **Example:** `RouterE(config)# router bgp 64502`	Enters router configuration mode for the specified routing process.
Step 16	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*] **Example:** `RouterE(config-router)# neighbor 192.168.3.1 remote-as 64503`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • In this example, the interface (192.168.3.1 in the figure above) at Router E is with the subnet range set for the BGP listen range group, group192. When TCP opens a session to peer to Router B, Router B creates this peer dynamically.
Step 17	**address-family l2vpn evpn** **Example:** `RouterE(config-router)# address-family l2vpn evpn`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 18	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `RouterE(config-router-af)# neighbor group192 activate`	Activates the neighbor or listen range peer group for the configured address family. • In this example, the neighbor 192.168.1.1 is activated for the L2VPN EVPN address family. **Note** Usually, BGP peer groups cannot be activated using this command, but the listen range peer groups are a special case.
Step 19	**end** **Example:** `RouterE(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 20	Return to the first router.	—
Step 21	**show ip bgp l2vpn evpn summary** **Example:** `RouterB# show ip bgp l2vpn evpn summary`	(Optional) Displays the BGP path, prefix, and attribute information for all connections to BGP neighbors. • In this step, the configuration has returned to Router A.

	Command or Action	Purpose
Step 22	**show ip bgp l2vpn evpn peer-group** [*peer-group-name*] [**summary**] **Example:** `RouterB# show ip bgp peer-group group192`	(Optional) Displays information about BGP peer groups.
Step 23	**show ip bgp l2vpn evpn neighbors** [*ip-address*] **Example:** `RouterB# show ip bgp l2vpn evpn neighbors 192.168.3.2`	(Optional) Displays information about BGP and TCP connections to neighbors. • In this example, information is displayed about the dynamically created neighbor at 192.168.3.2. The IP address of this BGP neighbor can be found in the output of either the **show ip bgp l2vpn evpnsummary** or the **show ipbgp l2vpn evpnpeer-group** command. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Verifying BGP Dynamic Neighbor Support for L2VPN EVPN address family

Use the **show running-config | section router bgp** command to view the configuration for L2VPN EVPN address family.

```
RouterB# show running-config | section router bgp
router bgp 64503
 bgp log-neighbor-changes
 bgp listen range 192.168.0.0/16 peer-group group192
 bgp listen range 172.0.0.0/8 peer-group group172
 bgp listen range ABCD::/64 peer-group v6group
 bgp listen limit 200
 no bgp default ipv4-unicast
 neighbor group172 peer-group
 neighbor group172 remote-as 64503
 neighbor group192 peer-group
 neighbor group192 remote-as 64501 alternate-as 64502
 neighbor v6group peer-group
 neighbor v6group remote-as 64502
 !
 address-family ipv4
 exit-address-family
 address-family l2vpn evpn
  neighbor group172 activate
  neighbor group172 send-community both
  neighbor group192 activate
  neighbor group192 send-community both
  neighbor v6group activate
  neighbor v6group send-community extended
 exit-address-family
```

After both Router B and Router E are configured, use the **show ipbgp l2vpn evpnsummary** command on Router B to view the regular BGP neighbor, 172.21.1.2, and the two BGP neighbors

that were created dynamically when Router A and Router E initiated TCP sessions for BGP peering to Router B. The output also shows information about the configured listen range subnet groups.

```
RouterB# sh ip bgp l2vpn evpn sum
BGP router identifier 192.168.0.1, local AS number 64503
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
*ABCD::2        4        64502       4       4        1    0    0 00:00:32        0
*172.0.0.2      4        64503       9       9        1    0    0 00:04:29        0
*192.168.0.2    4        64501       8       7        1    0    0 00:04:31        0
*192.168.0.3    4        64502       7       9        1    0    0 00:04:33        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 4, Subnet ranges: 3

BGP peergroup group172 listen range group members:
  172.0.0.0/8
BGP peergroup group192 listen range group members:
  192.168.0.0/16
BGP peergroup v6group listen range group members:
  ABCD::/64
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
Total dynamically created neighbors: 4/(200 max), Subnet ranges: 3
```

The following output from the **show ipbgp all summary** command shows that the BGP neighbor 192.168.3.2 was dynamically created and is a member of the listen range group, group192. The output also shows that the IP prefix range of 192.168.0.0/16 is defined for the listen range named group192. Similarly, the same is seen for the IPv6 neighbor range group, v6group.

```
RouterB# sh ip bgp all sum
For address family: L2VPN E-VPN
BGP router identifier 192.168.0.1, local AS number 64503
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
*ABCD::2        4        64502       4       4        1    0    0 00:00:03        0
*172.0.0.2      4        64503       8       8        1    0    0 00:04:00        0
*192.168.0.2    4        64501       8       6        1    0    0 00:04:02        0
*192.168.0.3    4        64502       7       8        1    0    0 00:04:05        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 4, Subnet ranges: 3
BGP peergroup group172 listen range group members:
  172.0.0.0/8
BGP peergroup group192 listen range group members:
  192.168.0.0/16
BGP peergroup v6group listen range group members:
  ABCD::/64
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd

Total dynamically created neighbors: 4/(200 max), Subnet ranges: 3
```

Configuring BGP IPv6 Dynamic Neighbor Support with VRF Support

In Cisco IOS XE Denali 16.3 release, support for BGP dynamic neighbors was extended to IPv6 BGP peering.

Note You can also configure BGP IPv6 dynamic neighbors without VRF support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*]
5. **address-family** [**ipv4** | **ipv6**] [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*]]
6. **bgp listen** [**limit** *max-number*]
7. **neighbor** *peer-group-name* **peer-group**
8. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*]
9. **address-family** [**ipv4** | **ipv6**] [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*]]
10. **neighbor** *peer-group-name* **activate**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted. • The configuration is entered on router B.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 64503`	Enters router configuration mode for the specified routing process.
Step 4	**bgp listen** [**limit** *max-number* | **range** *network* / *length* **peer-group** *peer-group-name*] **Example:** `Device(config-router)# bgp listen range 2001::0/64 peer-group group192`	Associates a subnet range with a BGP peer group and activates the BGP dynamic neighbors feature. • Use the optional **limit** keyword and *max-number* argument to define the maximum number of BGP dynamic neighbors that can be created. • Use the optional **range** keyword and *network* / *length* argument to define a prefix range to be associated with the specified peer group.

	Command or Action	Purpose
		• In this example, the prefix range 2001::0/64 is associated with the listen range group named group192.
Step 5	**address-family** [**ipv4** | **ipv6**] [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*]] **Example:** `Device(config-router-af)# address-family ipv6 unicast vrf vrf1`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 6	**bgp listen** [**limit** *max-number*] **Example:** `Device(config-router)# bgp listen limit 500`	Specifies the maximum number of prefixes in VRF address family.
Step 7	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router)# neighbor group192 peer-group`	Creates a BGP peer group. • In this example, a peer group named group192 is created. This group will be used as a listen range group.
Step 8	**neighbor** *peer-group-name* **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number...*] **Example:** `Device(config-router)# neighbor group192 remote-as 101 alternate-as 102`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv6 BGP neighbor table. • Use the optional **alternate-as** keyword and *autonomous-system-number* argument to identify up to five alternate autonomous system numbers for listen range neighbors. • In this example, the peer group named group192 is configured with two possible autonomous system numbers. **Note** The **alternate-as** keyword is used only with the listen range peer groups, not with individual BGP neighbors.
Step 9	**address-family** [**ipv4** | **ipv6**] [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*]] **Example:** `Device(config-router-af)# address-family ipv4 unicast vrf vrf1`	Enable IPv4 address family for this peer-group.
Step 10	**neighbor** *peer-group-name* **activate** **Example:**	Activates the neighbor or listen range peer group for the configured address family.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor group192 activate`	
Step 11	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Verifying BGP IPv6 Dynamic Neighbor Configuration

Use the **show ip bgp ipv6 unicast summary** command to verify the BGP IPv6 unicast address family configuration in global routing table:

```
Device# show ip bgp ipv6 unicast summary
BGP router identifier 192.168.3.1, local AS number 64503
BGP table version is 1, main routing table version 1
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
*2001::1 4 64502 2 2 0 0 0 00:00:37 0
* Dynamically created based on a listen range command
Dynamically created neighbors: 1/(200 max), Subnet ranges: 1
BGP peergroup group192 listen range group members:
2001::0/64
```

Use the **show ip bgp { ipv4 | ipv6 } unicast peer-group< name>** command to verify the IPv6 dynamic neighbors configuration in global routing table:

```
Device# show ip bgp ipv6 unicast peer-group group192
BGP peer-group is group192, remote AS 64501
BGP peergroup group192 listen range group members:
2001::0/64
BGP version 4
Default minimum time between advertisement runs is 30 seconds
For address family: IPv6 Unicast
BGP neighbor is group192, peer-group external, members:
*2001::1
Index 0, Offset 0, Mask 0x0
Update messages formatted 0, replicated 0
Number of NLRIs in the update sent: max 0, min 0
```

You can use the following commands to verify the BGP IPv6 dynamic neighbors configuration in the VRF routing table:

- **show ip bgp vpnv6 unicast vrf <name> neighbors**
- **show ip bgp vpnv6 unicast vrf <name> summary**
- **show ip bgp vpnv6 unicast vrf <name> peer-group <name>**
- **debug bgp [ipv6 | vpnv6] unicast range**

Block BGP Dynamic Neighbor Session Establishment with a Node

Usage Notes

- After you block BGP dynamic neighbor sessions to a node, the router rejects requests to create BGP dynamic neighbor sessions to the node.
- You can configure multiple block commands at the router level.
- The block command does not affect static BGP neighbor sessions.
- The router does not verify whether the IP address specified with the block command falls in the IP address range of the dynamic peer group.

To block a router from establishing a BGP dynamic neighbor session with a node, use the router-level command **bgp listen block** {*ipv4-address*|*ipv6-address*}.

```
router bgp 1
 bgp listen block ipv4-address
 bgp listen range subnet-ipv4-prefix/subnet-mask-length peer-group DYN_NBR_GROUP
 neighbor DYN_NBR_GROUP peer-group
 neighbor DYN_NBR_GROUP remote-as 200
 !
 address-family ipv4
  neighbor DYN_NBR_GROUP activate
 exit-address-family
!
```

You can use the **bgp listen block** {*ipv4-address*|*ipv6-address*} command to block dynamic neighbor sessions to global and VRF neighbors. To block dynamic neighbor sessions to VRF neighbors, use the command in the address family configuration mode.

Example:

```
vrf definition example
 rd 1:1
 address-family ipv4
  route-target export 1:1
  route-target import 1:1

router bgp 100
 bgp listen range 10.0.101.0/24 peer-group dn-group-v4
 address-family ipv4 vrf example
  bgp listen block 10.0.101.103
  bgp listen block 10.0.101.106
  neighbor dn-group-v4 peer-group
  neighbor dn-group-v4 remote-as 1.101
  neighbor dn-group-v4 activate
 exit-address-family
```

To undo the blocking of BGP dynamic neighbor sessions to a node, use the command **no bgp listen block** {*ipv4-address*|*ipv6-address*} at the router-level or in the address family configuration mode.

Related Topics

Block BGP Dynamic Neighbor Sessions, on page 902

View Blocked BGP Dynamic Neighbor Sessions

Use the **show ip bgp summary** command to view blocked BGP dynamic neighbor sessions.

```
Router#show ip bgp all summary
For address family: IPv4 Unicast
BGP router identifier 10.16.16.100, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
*10.16.16.2     4          200      40      39        1    0    0 00:34:07        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 1, Subnet ranges: 1

BGP peergroup DYN_NBR_GROUP listen range group members:
  10.16.16.0/24

Blocked Dynamic sessions:
  10.16.16.1
```

Debug Blocked BGP Dynamic Neighbor Sessions

The following debugging events related to blocked BGP dynamic neighbor sessions are added to the output of the debug command **debug ip bgp range** [**detail**]:

- Neighbor processing due to configuration of the block command or undoing the block configuration.
- BGP sessions that are not formed because of the block configuration.

Configuration Examples for BGP Dynamic Neighbors

Example: Implementing BGP Dynamic Neighbors Using Subnet Ranges

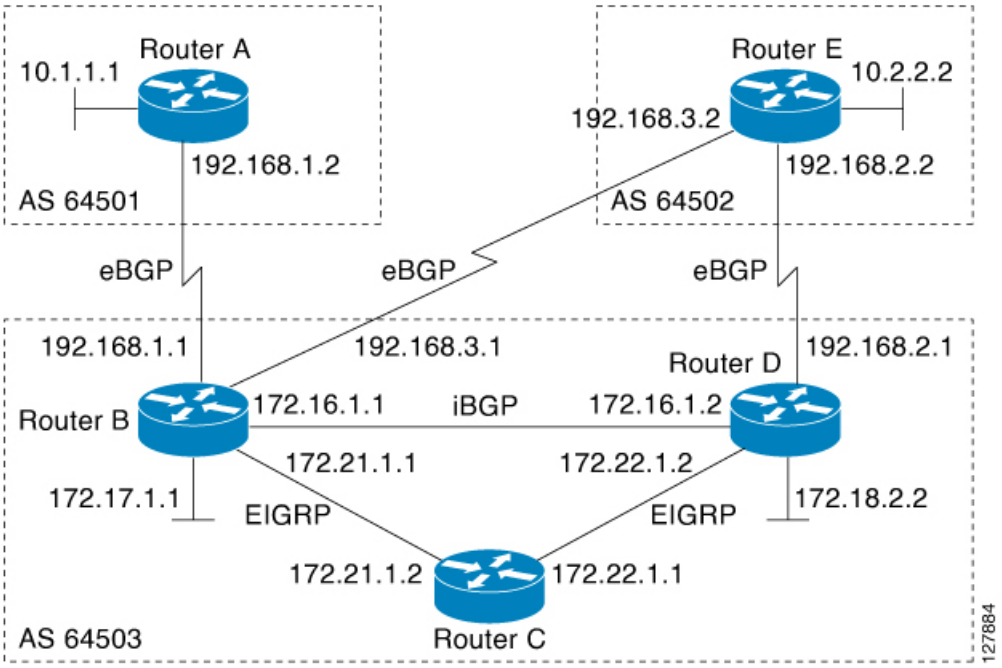
In the following example, two BGP peer groups are created on Router B in the figure below, a global limit is set on the number of dynamic BGP neighbors, and a subnet range is associated with a peer group. Configuring the subnet range enables the dynamic BGP neighbor process. The peer groups are added to the BGP neighbor table of the local router, and an alternate autonomous system number is also configured for one of the peer groups, group192. The subnet range peer groups and a standard BGP peer are then activated under the IPv4 address family.

The configuration moves to another router—Router A in the figure below—where a BGP session is started and the neighbor router, Router B, is configured as a remote BGP peer. The peering configuration opens a TCP session and triggers Router B to create a dynamic BGP neighbor because the IP address that starts the TCP session (192.168.1.2) is within the configured subnet range for dynamic BGP peers.

A third router—Router E in the figure below—also starts a BGP peering session with Router B. Router E is in the autonomous system 64502, which is the configured alternate autonomous system. Router B responds to the resulting TCP session by creating another dynamic BGP peer.

This example concludes with the output of the **show ip bgp summary** command entered on Router B.

Figure 68: BGP Dynamic Neighbor Topology



Router B

```
enable
configure terminal
router bgp 64503
 bgp log-neighbor-changes
 bgp listen limit 200
 bgp listen range 172.21.0.0/16 peer-group group172
 bgp listen range 192.168.0.0/16 peer-group group192
 neighbor group172 peer-group
 neighbor group172 remote-as 64503
 neighbor group192 peer-group
 neighbor group192 remote-as 64501 alternate-as 64502
 neighbor 172.16.1.2 remote-as 64503
 address-family ipv4 unicast
 neighbor group172 activate
 neighbor group192 activate
 neighbor 172.16.1.2 activate
 end
```

Router A

```
enable
configure terminal
router bgp 64501
 neighbor 192.168.1.1 remote-as 64503
 exit
```

Router E

```
enable
configure terminal
router bgp 64502
 neighbor 192.168.3.1 remote-as 64503
 exit
```

After both Router A and Router E are configured, the **show ip bgp summary** command is run on Router B. The output displays the regular BGP neighbor, 172.16.1.2, and the two BGP neighbors that were created dynamically when Router A and Router E initiated TCP sessions for BGP peering to Router B. The output also shows information about the configured listen range subnet groups.

```
BGP router identifier 192.168.3.1, local AS number 64503
BGP table version is 1, main routing table version 1
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.1.2      4 64503      15      15        1    0    0 00:12:20        0
*192.168.1.2    4 64501       3       3        1    0    0 00:00:37        0
*192.168.3.2    4 64502       6       6        1    0    0 00:04:36        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 2/(200 max), Subnet ranges: 2
BGP peergroup group172 listen range group members:
  172.21.0.0/16
BGP peergroup group192 listen range group members:
  192.168.0.0/16
```

Example: Configuring BGP IPv6 Dynamic Neighbor Support with VRF Support

Configuring BGP IPv6 Dynamic Neighbor Support with VRF Support

```
enable
configure terminal
router bgp 55000
  bgp listen range 2001::0/64 peer-group group182
  bgp listen limit 600
  address-family ipv6 unicast vrf vrf2
    bgp listen limit 600
    neighbor group182 peer-group
    neighbor group182 remote-as 103 alternate-as 104
  exit-address-family
  address-family ipv4 unicast vrf vrf2
    neighbor group182 activate
  exit-address-family
    end
```

Configuring BGP IPv6 Dynamic Neighbor Support without VRF Support

```
enable
configure terminal
router bgp 100
 bgp listen range 2001::0/64 peer-group group192
 bgp listen limit 500
 neighbor group192 peer-group
 neighbor group192 remote-as 64510 alternate-as 65511
 address family ipv6 unicast
  neighbor group192 activate
  address family ipv4 unicast
```

```
 neighbor group192 activate
end
```

Persistent Dynamic Neighbors

Persistent Dynamic Neighbor is a feature enhancement that will delay the deletion of dynamic neighbors even after the session is terminated. This feature prevents you from deleting the configured neighbors for a specified time or indefinitely after leaving the established state and therefore maintain the session information. The feature can be configured both globally and per peer-group. If the persistent feature is configured without a timer value, any dynamic neighbor associated with the configuration will be persistent indefinitely.

This functionality can also prove useful in other interoperability aspects like maximum-prefix and Non Stop Forwarding (NSF) that require maintaining the neighbor information after the session is no longer established.

Note Note: If the persistent feature is configured without a timer value, any dynamic neighbor associated with the configuration will be persistent indefinitely. The persistent dynamic neighbor timer must be larger than the maximum-prefix restart timer when configured together. Similarly, restart timer can be of any value if the Persistent timer is indefinite. For more information on BGP Maximum Prefix see, BGP Maximum Prefix on IOS XE

How to configure Persistent Dynamic Neighbors

Configuring Persistent Dynamic Neighbor

Perform this task to configure Persistent Dynamic Neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp listen range** <*range*> **peer-group** <*pg*> [**persistent** [<*1-65535*>]]
5. **bgp listen** [**persistent** [<*1-65535*>]]
6. **neighbor** *peer-group-name* **peer-group**
7. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number*
8. **address-family** *address-family*
9. **neighbor** { *ip-address* \ *peer-group-name*} **activate**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters Global Configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 3`	Specifies the autonomous system number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 4	**bgp listen range** <*range*> **peer-group** <*pg*> [**persistent** [<*1-65535*>]] **Example:** `Device(config-bgp)# bgp listen range 1.1.1.0/24 peer-group DN persistent 60`	**Note** Please use the **bgp listen range peer group persistent** command to enable Persistent Dynamic Neighbor per range group. Specifies the time for the Persistent Dynamic timer. **Note** If no timer value is provided, dynamic neighbors will be persistent indefinitely.
Step 5	**bgp listen** [**persistent** [<*1-65535*>]] **Example:** `Device(config-bgp)# bgp listen range 1.1.1.0/24 peer-group DN persistent 60`	**Note** Please use the **bgp listen persistent** command to enable Persistent Dynamic Neighbor globally. Specifies the time for the Persistent Dynamic timer.
Step 6	**neighbor** *peer-group-name* **peer-group** **Example:** `Device (config-bgp)# neighbor DN peer-group`	Creates a BGP peer group.
Step 7	**neighbor** *peer-group-name* **remote-as** *autonomous-system-number* **Example:** `Device (config-bgp)# neighbor DN remote-as 1000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the BGP neighbor table of the local router.
Step 8	**address-family** *address-family* **Example:** `Device(config-bgp)# addressfamily ipv4 unicast`	Enable IPv4 address family for this peer-group and enters address family configuration submode.
Step 9	**neighbor** { *ip-address* \ *peer-group-name*} **activate** **Example:** `(config-bgp-nbr-af)# neighbor DN activate`	Activates the neighbor or listen range peer group for the configured address family.
Step 10	**end** **Example:** `Device(config-bgp-nbr-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Example for Persistent Dynamic Neighbor

The following example shows how to configure BGP Persistent Dynamic Neighbor feature for the IPv4 address family:

```
Router bgp 3
bgp listen range 1.1.1.0/24 peer-group DN persistent 60
neighbor DN peer-group
neighbor DN remote-as 1000
address-family ipv4 unicast
neighbor DN activate
```

Troubleshooting

The following output from the **show ip bgp [address-family] summary** command shows the added additional information about the number of persistent dynamic neighbors

```
Router# show ip bgp ipv4 unicast summary
BGP router identifier 10.0.96.1, local AS number 100
BGP table version is 1, main routing table version 1
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
*10.0.101.1 4 1 3 4 1 0 0 00:00:19 0
* Dynamically created based on a listen range command
Dynamically created neighbors: 1 (1 persistent), Subnet ranges: 4
BGP peergroup DN1 listen range group members:
10.0.0.0/16
Number of dynamically created neighbors in vrf red: 2/(200 max)
Total dynamically created neighbors: 4/(400 max) (4 persistent), Subnet ranges: 4
```

The following output from the **show ip bgp neighbor A.B.C.D** command shows the added additional information about if/when the neighbor will be deleted.

```
Router# show ip bgp neighbors 10.0.101.1
BGP neighbor is *10.0.101.1, remote AS 1, external link
Member of peer-group DN1 for session parameters
Belongs to the subnet range group: 10.0.0.0/16
BGP version 4, remote router ID 0.0.0.0
BGP state = Idle, down for 00:00:02
Persistent Dynamic Neighbor:
persistence timer: 1
deleting in: 1 minutes
Last update received: n/a
...
```

The following output from the **clear ip bgp X.X.X.X** command shows that persistent dynamic neighbors can be cleaned.

In the following example, a soft reconfiguration is initiated for the inbound session with the neighbor 10.100.0.1, and the outbound session is unaffected:

```
Router# clear ip bgp 10.100.0.1 soft in
```

In the following example, the route refresh capability is enabled on the BGP neighbor routers and a soft reconfiguration is initiated for the inbound session with the neighbor 172.16.10.2, and the outbound session is unaffected:

```
Router# clear ip bgp 172.16.10.2 in
```

In the following example, a hard reset is initiated for sessions with all routers in the autonomous system numbered 35700:

```
Router# clear ip bgp 35700
```

In the following example, a hard reset is initiated for sessions with all routers in the 4-byte autonomous system numbered 65538 in asplain notation. This example requires Cisco IOS Release 12.0(32)SY8, 12.0(33)S3, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, Cisco IOS XE Release 2.4, or a later release.

```
Router# clear ip bgp 65538
```

In the following example, a hard reset is initiated for sessions with all routers in the 4-byte autonomous system numbered 1.2 in asdot notation. This example requires Cisco IOS Release 12.0(32)SY8, 12.0(32)S12, 12.2(33)SRE, 12.2(33)XNE, 12.2(33)SXI1, 12.4(24)T, and Cisco IOS XE Release 2.3, or a later release.

```
Router# clear ip bgp 1.2
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Dynamic Neighbors

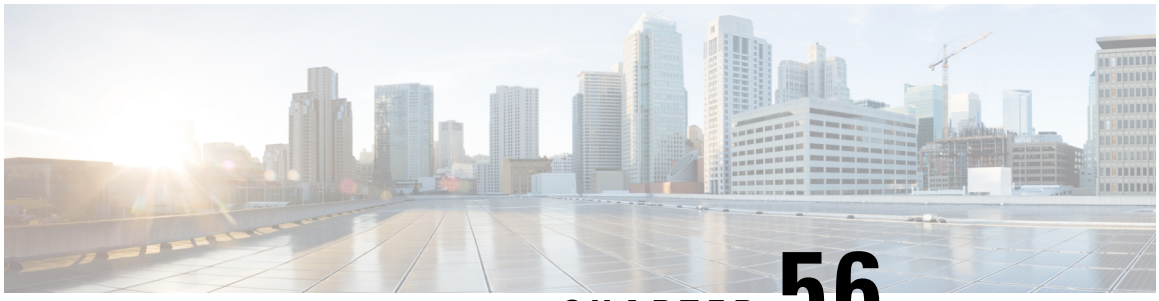
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 81: Feature Information for BGP Dynamic Neighbors

Feature Name	Releases	Feature Information
BGP Dynamic Neighbors		BGP dynamic neighbor support allows BGP peering to a group of remote neighbors that are defined by a range of IP addresses. Each range can be configured as a subnet IP address. BGP dynamic neighbors are configured using a range of IP addresses and BGP peer groups. After a subnet range is configured for a BGP peer group and a TCP session is initiated for an IP address in the subnet range, a new BGP neighbor is dynamically created as a member of that group. The new BGP neighbor will inherit any configuration for the peer group. The following commands were introduced or modified by this feature: **bgp listen**, **debug ip bgp range**, **neighbor remote-as**, **show ip bgp neighbors**, **show ip bgp peer-group**, and **show ip bgp summary**.
BGP IPv6 Dynamic Neighbor Support and VRF Support	Cisco IOS XE Denali 16.3.1	In Cisco IOS XE Denali 16.3 release, support for BGP dynamic neighbors was extended to IPv6 BGP peering with support for VRF. The following commands were introduced or modified by this feature: **bgp listen**, **debug ip bgp range**, **neighbor remote-as**, **show bgp neighbors**, **show bgp summary**, **show bgp vpnv6 unicast vrf neighbors**, **show bgp vpnv6 unicast vrf peer-group** , **show bgp vpnv6 unicast vrf summary**.

Feature Name	Releases	Feature Information
Block BGP Dynamic Neighbor Sessions	Cisco IOS XE Amsterdam 17.2.1	From IOS XE Release 17.2.1, you can block a router from forming BGP dynamic neighbor sessions with certain nodes in a BGP peer group by identifying these nodes by their IP addresses. The following commands are introduced or modified: **bgp listen block** {*ipv4-address*|*ipv6-address*}, **show ip bgp summary**, **debug ip bgp range** [**detail**].
BGP Dynamic Neighbor Support for L2VPN EVPN and other address families	Cisco IOS XE Dublin 17.11.1a	From Cisco IOS XE Dublin 17.11.1a release, support for BGP dynamic neighbors is extended to the following address families: • Layer 2 VPN Ethernet VPN (EVPN) • Layer 2 VPN Virtual Private LAN Service (VPLS) • IPv4 FlowSpec • IPv4 MDT • IPv4 Multicast • IPv4 Multicast VPN (MVPN) • IPv6 FlowSpec • IPv6 Multicast • IPv6 Multicast VPN (MVPN) • Link-State • Network Service Access Point (NSAP) • RT-filter
Support for Persistence of BGP Dynamic Neighbors	Cisco IOX XE 17.13.1a	From IOS XE 17.13.1a, the device maintains the neighbor information even after the session is terminated. To configure this, use the **bgp listen persistent** command for all dynamic neighbors and **bgp listen range peer-group persistent** command for specific neighbors.



CHAPTER 56

BGP Support for Next-Hop Address Tracking

The BGP Support for Next-Hop Address Tracking feature is enabled by default when a supporting Cisco software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a bestpath calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

Information About BGP Support for Next-Hop Address Tracking

BGP Next-Hop Address Tracking

The BGP next-hop address tracking feature is enabled by default when a supporting Cisco software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a best-path calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

BGP Next-Hop Dampening Penalties

If the penalty threshold value is higher than 950, then the delay is calculated as the reuse time using the dampening calculations. The dampening calculations use the following parameters:

- Penalty
- Half-life time
- Reuse time
- max-suppress-time

The values for the dampening parameters used are a max-suppress-time of 60 seconds, the half-life of 8 seconds, and the reuse-limit of 100.

For example, if the original penalty of 1600 is added, then after 16 seconds it becomes 800, and after 40 seconds, the penalty becomes 100. Hence, for the route update penalty of 1600, a delay of 40 seconds is used to schedule the BGP scanner.

These parameters (penalty threshold and any of the dampening parameters) cannot be modified.

Default BGP Scanner Behavior

BGP monitors the next hop of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. By default, the BGP scanner is used to poll the RIB for this information every 60 seconds. During the 60 second time period between scan cycles, Interior Gateway Protocol (IGP) instability or other network failures can cause null routes and routing loops to temporarily form.

BGP Next_Hop Attribute

The Next_Hop attribute identifies the next-hop IP address to be used as the BGP next hop to the destination. The device makes a recursive lookup to find the BGP next hop in the routing table. In external BGP (eBGP), the next hop is the IP address of the peer that sent the update. Internal BGP (iBGP) sets the next-hop address to the IP address of the peer that advertised the prefix for routes that originate internally. When any routes to iBGP that are learned from eBGP are advertised, the Next_Hop attribute is unchanged.

A BGP next-hop IP address must be reachable in order for the device to use a BGP route. Reachability information is usually provided by the IGP, and changes in the IGP can influence the forwarding of the next-hop address over a network backbone.

Selective BGP Next-Hop Route Filtering

BGP selective next-hop route filtering was implemented as part of the BGP Selective Address Tracking feature to support BGP next-hop address tracking. Selective next-hop route filtering uses a route map to selectively define routes to help resolve the BGP next hop.

The ability to use a route map with the **bgp nexthop** command allows the configuration of the length of a prefix that applies to the BGP Next_Hop attribute. The route map is used during the BGP bestpath calculation and is applied to the route in the routing table that covers the next-hop attribute for BGP prefixes. If the next-hop route fails the route map evaluation, the next-hop route is marked as unreachable. This command is per address family, so different route maps can be applied for next-hop routes in different address families.

Note Use route map on ASR series devices to set the next hop as BGP peer for the route and apply that route map in outbound direction towards the peer.

Note Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

BGP Support for Fast Peering Session Deactivation

BGP Hold Timer

By default, the BGP hold timer is set to run every 180 seconds in Cisco software. This timer value is set as the default to protect the BGP routing process from instability that can be caused by peering sessions with other routing protocols. BGP devices typically carry large routing tables, so frequent session resets are not desirable.

BGP Fast Peering Session Deactivation

BGP fast peering session deactivation improves BGP convergence and response time to adjacency changes with BGP neighbors. This feature is event driven and configured on a per-neighbor basis. When this feature is enabled, BGP will monitor the peering session with the specified neighbor. Adjacency changes are detected and terminated peering sessions are deactivated in between the default or configured BGP scanning interval.

Selective Address Tracking for BGP Fast Session Deactivation

In Cisco IOS XE Release 2.1 and later releases, the BGP Selective Address Tracking feature introduced the use of a route map with BGP fast session deactivation. The **route-map** keyword and *map-name* argument are used with the **neighbor fall-over** BGP neighbor session command to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset. The route map is not used for session establishment.

Note Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

How to Configure BGP Support for Next-Hop Address Tracking

Configuring BGP Next-Hop Address Tracking

The tasks in this section show how configure BGP next-hop address tracking. BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP. For more details about configuring route dampening, see “Configuring BGP Route Dampening.”

Configuring BGP Selective Next-Hop Route Filtering

Perform this task to configure selective next-hop route filtering using a route map to filter potential next-hop routes. This task uses prefix lists and route maps to match IP addresses or source protocols and can be used to avoid aggregate addresses and BGP prefixes being considered as next-hop routes. Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

For more examples of how to use the **bgp nexthop** command, see the "Examples: Configuring BGP Selective Next-Hop Route Filtering" section in this module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast**]
5. **bgp nexthop route-map** *map-name*
6. **exit**
7. **exit**
8. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network / length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
9. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
10. **match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*]
11. **exit**
12. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
13. **end**
14. **show ip bgp** [*network*] [*network-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**unicast**	**multicast**] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command.

	Command or Action	Purpose
		• The **multicast** keyword specifies IPv4 multicast address prefixes.
Step 5	**bgp nexthop route-map** *map-name* **Example:** `Device(config-router-af)# bgp nexthop route-map CHECK-NEXTHOP`	Permits a route map to selectively define routes to help resolve the BGP next hop. • In this example the route map named CHECK-NEXTHOP is created.
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 8	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network / length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Device(config)# ip prefix-list FILTER25 seq 5 permit 0.0.0.0/0 le 25`	Creates a prefix list for BGP next-hop route filtering. • Selective next-hop route filtering supports prefix length matching or source protocol matching on a per address-family basis. • The example creates a prefix list named FILTER25 that permits routes only if the mask length is more than 25; this will avoid aggregate routes being considered as the next-hop route.
Step 9	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map CHECK-NEXTHOP deny 10`	Configures a route map and enters route map configuration mode. • In this example, a route map named CHECK-NEXTHOP is created. If there is an IP address match in the following **match** command, the IP address will be denied.
Step 10	**match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*] **Example:** `Device(config-route-map)# match ip address prefix-list FILTER25`	Matches the IP addresses in the specified prefix list. • Use the *prefix-list-name* argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route map configuration mode and enters global configuration mode.
Step 12	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map CHECK-NEXTHOP permit 20`	Configures a route map and enters route map configuration mode. • In this example, all other IP addresses are permitted by route map CHECK-NEXTHOP.
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.
Step 14	**show ip bgp** [*network*] [*network-mask*] **Example:** `Device# show ip bgp`	Displays the entries in the BGP routing table. • Enter this command to view the next-hop addresses for each route. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

Example

The following example from the **show ip bgp** command shows the next-hop addresses for each route:

```
BGP table version is 7, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*  10.1.1.0/24      192.168.1.2              0             0 40000 i
*  10.2.2.0/24      192.168.3.2              0             0 50000 i
*> 172.16.1.0/24    0.0.0.0                  0         32768 i
*> 172.17.1.0/24    0.0.0.0                  0         32768
```

Adjusting the Delay Interval for BGP Next-Hop Address Tracking

Perform this task to adjust the delay interval between routing table walks for BGP next-hop address tracking.

You can increase the performance of this feature by tuning the delay interval between full routing table walks to match the tuning parameters for the Interior Gateway protocol (IGP). The default delay interval is 5 seconds. This value is optimal for a fast-tuned IGP. In the case of an IGP that converges more slowly, you can change the delay interval to 20 seconds or more, depending on the IGP convergence time.

BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor

sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [[**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]]
5. **bgp nexthop trigger delay** *delay-timer*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 64512`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**address-family ipv4** [[**mdt**	**multicast**	**tunnel**	**unicast** [**vrf** *vrf-name*]	**vrf** *vrf-name*]	**vpnv4** [**unicast**]] **Example:** `Device(config-router)# address-family ipv4 unicast`	Enter address family configuration mode to configure BGP peers to accept address family-specific configurations. • The example creates an IPv4 unicast address family session.
Step 5	**bgp nexthop trigger delay** *delay-timer* **Example:** `Device(config-router-af)# bgp nexthop trigger delay 20`	Configures the delay interval between routing table walks for next-hop address tracking. • The time period determines how long BGP will wait before starting a full routing table walk after notification is received. • The value for the *delay-timer* argument is a number from 1 to 100 seconds. The default value is 5 seconds. • The example configures a delay interval of 20 seconds.
Step 6	**end** **Example:**	Exits address-family configuration mode, and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router-af)# end`	

Disabling BGP Next-Hop Address Tracking

Perform this task to disable BGP next-hop address tracking. BGP next-hop address tracking is enabled by default under the IPv4 and VPNv4 address families. Beginning with Cisco IOS Release 12.2(33)SB6, BGP next-hop address tracking is also enabled by default under the VPNv6 address family whenever the next hop is an IPv4 address mapped to an IPv6 next-hop address.

Disabling next hop address tracking may be useful if you the network has unstable IGP peers and route dampening is not resolving the stability issues. To reenable BGP next-hop address tracking, use the **bgp nexthop**command with the **trigger** and **enable** keywords.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [[**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**] | **vpnv6** [**unicast**]]
5. **no bgp nexthop trigger enable**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 64512`	Enters router configuration mod to create or configure a BGP routing process.
Step 4	**address-family ipv4** [[**mdt**	**multicast**	**tunnel**	**unicast** [**vrf** *vrf-name*]	**vrf** *vrf-name*]	**vpnv4** [**unicast**]	**vpnv6** [**unicast**]] **Example:** `Device(config-router)# address-family ipv4 unicast`	Enter address family configuration mode to configure BGP peers to accept address family-specific configurations. • The example creates an IPv4 unicast address family session.

	Command or Action	Purpose
Step 5	**no bgp nexthop trigger enable** **Example:** `Device(config-router-af)# no bgp nexthop trigger enable`	Disables BGP next-hop address tracking. • Next-hop address tracking is enabled by default for IPv4 and VPNv4 address family sessions. • The example disables next-hop address tracking.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address-family configuration mode, and enters Privileged EXEC mode.

Configuring Fast Session Deactivation

The tasks in this section show how to configure BGP next-hop address tracking. BGP next-hop address tracking significantly improves the response time of BGP to next-hop changes in the RIB. However, unstable Interior Gateway Protocol (IGP) peers can introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP. For more details about route dampening, see the "Configuring Internal BGP Features" module.

Configuring Fast Session Deactivation for a BGP Neighbor

Perform this task to establish a peering session with a BGP neighbor and then configure the peering session for fast session deactivation to improve the network convergence time if the peering session is deactivated.

Enabling fast session deactivation for a BGP neighbor can significantly improve BGP convergence time. However, unstable IGP peers can still introduce instability to BGP neighbor sessions. We recommend that you aggressively dampen unstable IGP peering sessions to reduce the possible impact to BGP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **fall-over**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations. • The example creates an IPv4 unicast address family session.
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 10.0.0.1 remote-as 50000`	Establishes a peering session with a BGP neighbor.
Step 6	**neighbor** *ip-address* **fall-over** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 fall-over`	Configures the BGP peering to use fast session deactivation. • BGP will remove all routes learned through this peer if the session is deactivated.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits configuration mode and returns to privileged EXEC mode.

Configuring Selective Address Tracking for Fast Session Deactivation

Perform this task to configure selective address tracking for fast session deactivation. The optional **route-map** keyword and *map-name* argument of the **neighbor fall-over** command are used to determine if a peering session with a BGP neighbor should be deactivated (reset) when a route to the BGP peer changes. The route map is evaluated against the new route, and if a deny statement is returned, the peer session is reset.

Note Only **match ip address** and **match source-protocol** commands are supported in the route map. No **set** commands or other **match** commands are supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **fall-over** [**route-map** *map-name*]
6. **exit**
7. **ip prefix-list** *list-name* [**seq** *seq-value*]{**deny** *network / length* | **permit** *network / length*}[**ge** *ge-value*] [**le** *le-value*]
8. **route-map** *map-name* [**permit** | **deny**][*sequence-number*]
9. **match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 5	**neighbor** *ip-address* **fall-over** [**route-map** *map-name*] **Example:** `Device(config-router)# neighbor 192.168.1.2 fall-over route-map CHECK-NBR`	Applies a route map when a route to the BGP changes. • In this example, the route map named CHECK-NBR is applied when the route to neighbor 192.168.1.2 changes.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**ip prefix-list** *list-name* [**seq** *seq-value*]{**deny** *network* / *length* | **permit** *network* / *length*}[**ge** *ge-value*] [**le** *le-value*] **Example:** `Device(config)# ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28`	Creates a prefix list for BGP next-hop route filtering. • Selective next-hop route filtering supports prefix length matching or source protocol matching on a per-address-family basis. • The example creates a prefix list named FILTER28 that permits routes only if the mask length is greater than or equal to 28.
Step 8	**route-map** *map-name* [**permit** | **deny**][*sequence-number*] **Example:** `Device(config)# route-map CHECK-NBR permit 10`	Configures a route map and enters route-map configuration mode. • In this example, a route map named CHECK-NBR is created. If there is an IP address match in the following **match** command, the IP address will be permitted.
Step 9	**match ip address prefix-list** *prefix-list-name* [*prefix-list-name...*] **Example:** `Device(config-route-map)# match ip address prefix-list FILTER28`	Matches the IP addresses in the specified prefix list. • Use the *prefix-list-name* argument to specify the name of a prefix list. The ellipsis means that more than one prefix list can be specified. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 10	**end** **Example:** `Device(config-route-map)# end`	Exits configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP Support for Next-Hop Address Tracking

Example: Enabling and Disabling BGP Next-Hop Address Tracking

In the following example, next-hop address tracking is disabled under the IPv4 address family session:

```
router bgp 50000
 address-family ipv4 unicast
 no bgp nexthop trigger enable
```

Example: Adjusting the Delay Interval for BGP Next-Hop Address Tracking

In the following example, the delay interval for next-hop tracking is configured to occur every 20 seconds under the IPv4 address family session:

```
router bgp 50000
 address-family ipv4 unicast
 bgp nexthop trigger delay 20
```

Examples: Configuring BGP Selective Next-Hop Route Filtering

The following example shows how to configure BGP selective next-hop route filtering to avoid using a BGP prefix as the next-hop route. If the most specific route that covers the next hop is a BGP route, then the BGP route will be marked as unreachable. The next hop must be an IGP or static route.

```
router bgp 45000
 address-family ipv4 unicast
 bgp nexthop route-map CHECK-BGP
 exit
 exit
route-map CHECK-BGP deny 10
 match source-protocol bgp 1
 exit
route-map CHECK-BGP permit 20
 end
```

The following example shows how to configure BGP selective next-hop route filtering to avoid using a BGP prefix as the next-hop route and to ensure that the prefix is more specific than /25.

```
router bgp 45000
 address-family ipv4 unicast
 bgp nexthop route-map CHECK-BGP25
 exit
 exit
ip prefix-list FILTER25 seq 5 permit 0.0.0.0/0 le 25
route-map CHECK-BGP25 deny 10
 match ip address prefix-list FILTER25
 exit
route-map CHECK-BGP25 deny 20
 match source-protocol bgp 1
 exit
route-map CHECK-BGP25 permit 30
 end
```

Example: Configuring Fast Session Deactivation for a BGP Neighbor

In the following example, the BGP routing process is configured on device A and device B to monitor and use fast peering session deactivation for the neighbor session between the two devices. Although fast peering session deactivation is not required at both devices in the neighbor session, it will help the BGP networks in both autonomous systems to converge faster if the neighbor session is deactivated.

Device A

```
router bgp 40000
 neighbor 192.168.1.1 remote-as 45000
```

```
 neighbor 192.168.1.1 fall-over
 end
```

Device B

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 fall-over
 end
```

Example: Configuring Selective Address Tracking for Fast Session Deactivation

The following example shows how to configure the BGP peering session to be reset if a route with a prefix of /28 or a more specific route to a peer destination is no longer available:

```
router bgp 45000
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 fall-over route-map CHECK-NBR
 exit
ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28
route-map CHECK-NBR permit 10
 match ip address prefix-list FILTER28
 end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

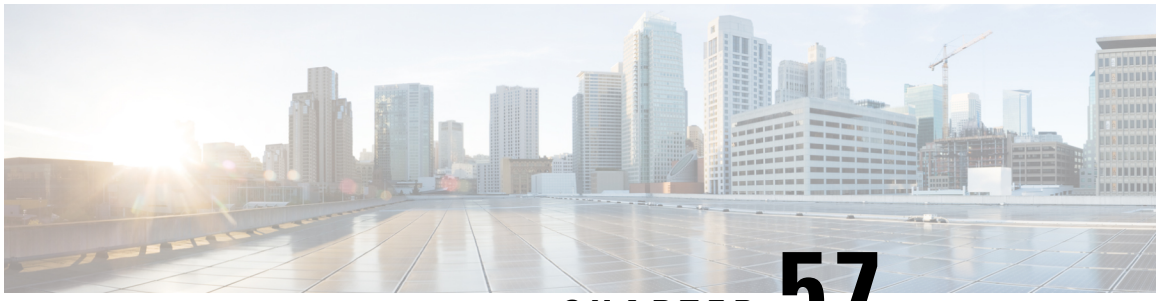
Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for Next-Hop Address Tracking

Table 82: Feature Information for BGP Support for Next-Hop Address Tracking

Feature Name	Releases	Feature Information
BGP Support for Next-Hop Address Tracking		The BGP Support for Next-Hop Address Tracking feature is enabled by default when a supporting Cisco IOS software image is installed. BGP next-hop address tracking is event driven. BGP prefixes are automatically tracked as peering sessions are established. Next-hop changes are rapidly reported to the BGP routing process as they are updated in the RIB. This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a bestpath calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed. The following command was introduced in this feature: **bgp nexthop**.
BGP Selective Address Tracking		The BGP Selective Address Tracking feature introduces the use of a route map for next-hop route filtering and fast session deactivation. Selective next-hop filtering uses a route map to selectively define routes to help resolve the BGP next hop, or a route map can be used to determine if a peering session with a BGP neighbor should be reset when a route to the BGP peer changes. The following commands were modified by this feature: **bgp nexthop**, **neighbor fall-over**.

Feature Name	Releases	Feature Information
BGP Support for Fast Peering Session Deactivation		The BGP Support for Fast Peering Session Deactivation feature introduced an event-driven notification system that allows a Border Gateway Protocol (BGP) process to monitor BGP peering sessions on a per-neighbor basis. This feature improves the response time of BGP to adjacency changes by allowing BGP to detect an adjacency change and deactivate the terminated session in between standard BGP scanning intervals. Enabling this feature improves overall BGP convergence. The following command was modified by this feature: **neighbor fall-over**.



CHAPTER 57

BGP Maximum-Prefix on IOS XE

This document provides information on the Border Gateway Protocol (BGP) Maximum - Prefix feature.

Information About Maximum-Prefix

IOS XE BGP maximum-prefix feature imposes a maximum limit on the number of prefixes that are received from a neighbor for a given address family. Whenever the number of prefixes received exceeds the maximum number configured, the BGP session is terminated, which is the default behavior, after sending a cease notification to the neighbor. The session is down until a manual clear is performed by the user. The session can be resumed by using the **clear bgp** command. It is possible to configure a period after which the session can be automatically brought up by using the **maximum prefix** command with the **restart** keyword. The maximum prefix limit can be configured by the user.

Note Maximum-Prefix feature on dynamic neighbors is only supported when the Persistent Dynamic Neighbors feature is configured. For more information refer Persistent Dynamic Neighbor

Maximum-Prefix logging events

In earlier versions of IOS-XE, the logging of maximum-prefix warnings was limited to one warning per log type within a 60-second time window, regardless of the specific neighbor triggering the warning. This means that if multiple neighbors exceeded the maximum-prefix limit within a short time frame, only the first warning was logged, and subsequent warnings were considered time-limited and not logged individually. From Cisco IOS XE 17.13.1a, the enhancement ensures the logging of maximum-prefix warnings are now time-limited per-neighbor within a 60-second time window.

BGP Maximum Prefix-Discard Extra

An option to discard extra is added to the maximum-prefix configuration. Configuring the discard extra option drops all excess prefixes received from the neighbor when the prefixes exceed the configured maximum value. This drop does not, however, result in session flap.

The benefits of discard extra option are:

- Limits the memory footstamp of BGP.
- Stops the flapping of the peer if the paths exceed the set limit.

On the same lines, the following describes the actions when the maximum prefix value is changed:

- If the maximum value alone is changed, a route-refresh message is sourced, if applicable.
- If the new maximum value is greater than the current prefix count state, the new prefix states are saved.
- If the new maximum value is less than the current prefix count state, then some existing prefixes are deleted to match the new configured state value.

Restrictions

- When the router drops prefixes, it is inconsistent with the rest of the network, resulting in possible routing loops.
- If prefixes are dropped, the standby and active BGP sessions may drop different prefixes. Consequently, an NSR switchover results in inconsistent BGP tables.
- The discard extra configuration cannot co-exist with the soft reconfig configuration.
- There is currently no way to control which prefixes are deleted.
- A peer may withdraw prefixes after some prefixes have been discarded. This may result in having discarded prefixes and still be below the prefix limit. To recover discarded prefixes up to the prefix limit, users may perform a soft clear on the neighbor.
- All maximum-prefix sub-options are mutually exclusive, only one can be configured at a time for a given neighbor.

Configuring Discard Extra

Perform this task to configure BGP maximum-prefix discard extra.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp***autonomous-system-number*
4. **neighbor***ip-address***remote-as***autonomous-system-number*
5. **address-family***address family*
6. **neighbor***ip-address***activate**
7. **neighbor***ip-address**maximum-prefix***prefix-limit** [*threshold*] [**discard-extra**] [**restart***minutes*] [**warning-only**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters Global Configuration mode.
Step 3	**router bgp***autonomous-system-number* **Example:** `Router(config)# router bgp 3`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor***ip-address***remote-as***autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.10.10.2 remote-as 2`	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 5	**address-family***address family* **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the address family and enters address family configuration submode.
Step 6	**neighbor***ip-address***activate** **Example:** `Router(config-router-af)# neighbor 10.10.10.2 activate`	Enables the neighbor to exchange prefixes for the given address family with the local router.

	Command or Action	Purpose
Step 7	**neighbor***ip-address**maximum-prefix***prefix***prefix-limit** [*threshold*] [**discard-extra**] [**restart***minutes*] [**warning-only**] **Example:** `Router(config-router-af)# neighbor 10.10.10.2 maximum-prefix 3 discard-extra`	Configures a limit to the number of prefixes allowed. Configures discard extra paths to discard extra paths when the maximum prefix limit is exceeded.
Step 8	**end**	Prompts user to take one of these actions: • **Yes** - Saves configuration changes and exits the configuration session. • **No** - Exits the configuration session without committing the configuration changes. • **Cancel** - Remains in the configuration session, without committing the configuration changes

Configuration Examples for Discard Extra

The following example shows how to configure BGP maximum-prefix discard extra feature for the IPv4 address family:

```
router bgp 3
neighbor 10.10.10.2 remote-as 2
address-family ipv4 unicast
neighbor 10.10.10.2 activate
neighbor 10.10.10.2 maximum-prefix 3 discard-extra
```

Verifying Discard Extra

The following **show ip bgp neighbor A.B.C.D** command displays the information about the number of prefixes that were discarded.

```
Device #show ip bgp neighbors 10.10.10.2
BGP neighbor is 10.10.10.2, remote AS 2, external link
…
For address family: IPv4 Unicast
Session: 10.10.10.2
BGP table version 1, neighbor version 1/0
Output queue size : 0
Index 3, Advertise bit 0
3 update-group member
Outbound path policy configured
Route map for outgoing advertisements is PEER_OUT
Slow-peer detection is disabled
Slow-peer split-update-group dynamic is disabled
Prefix activity: ---- ----
Prefixes Current: 0 4 (Consumes 544 bytes)
Prefixes Total: 0 4
Implicit Withdraw: 0 0
Explicit Withdraw: 0 0
Used as bestpath: n/a 0
```

```
Used as multipath: n/a 0
Used as secondary: n/a 0
                                Outbound Inbound
Local Policy Denied Prefixes: -------- -------
Total:                             0      1
Maximum prefixes allowed 3 (discard-extra)
Threshold for warning message 75%
Prefixes discarded: 1
```

Information About BGP Neighbor Session Restart After Max-Prefix Limit Reached

Prefix Limits and BGP Peering Sessions

Use the **neighbor maximum-prefix** command to limit the maximum number of prefixes that a device running BGP can receive from a peer. When the device receives too many prefixes from a peer and the maximum-prefix limit is exceeded, the peering session is disabled or brought down. The session stays down until the network operator manually brings the session back up by entering the **clear ip bgp** command, which clears stored prefixes.

BGP Neighbor Session Restart with the Maximum Prefix Limit

The **restart** keyword was added to the **neighbor maximum-prefix** command so that a network operator can configure a device to automatically reestablish a BGP neighbor peering session when the peering session has been disabled or brought down. The time interval at which peering can be reestablished automatically is configurable. The *restart-interval* for the **restart** keyword is specified in minutes; range is from 1 to 65,535 minutes.

Subcodes for BGP Cease Notification

Border Gateway Protocol (BGP) imposes maximum limits on the maximum number of prefixes that are accepted from a peer for a given address family. This limitation safeguards the device from resource depletion caused by misconfiguration, either locally or on the remote neighbor. To prevent a peer from flooding BGP with advertisements, a limit is placed on the number of prefixes that are accepted from a peer for each supported address family. The default limits can be overridden through configuration of the maximum-prefix limit command for the peer for the appropriate address family.

The following subcodes are supported for the BGP cease notification message:

- Maximum number of prefixes reached
- Administrative shutdown
- Peer de-configured
- Administrative reset

A cease notification message is sent to the neighbor and the peering with the neighbor is terminated when the number of prefixes received from the peer for a given address family exceeds the maximum limit (either set by default or configured by the user) for that address family. It is possible that the maximum number of

prefixes for a neighbor for a given address family has been configured after the peering with the neighbor has been established and a certain number of prefixes have already been received from the neighbor for that address family. A cease notification message is sent to the neighbor and peering with the neighbor is terminated immediately after the configuration if the configured maximum number of prefixes is fewer than the number of prefixes that have already been received from the neighbor for the address family.

How to Configure a Device to Reestablish a Neighbor Session After the Maximum Prefix Limit Has Been Exceeded

Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Reached

Perform this task to configure the time interval at which a BGP neighbor session is reestablished by a device when the number of prefixes that have been received from a BGP peer has exceeded the maximum prefix limit.

The network operator can configure a device running BGP to automatically reestablish a neighbor session that has been brought down because the configured maximum-prefix limit has been exceeded. No intervention from the network operator is required when this feature is enabled.

Note This task attempts to reestablish a disabled BGP neighbor session at the configured time interval that is specified by the network operator. However, the configuration of the restart timer alone cannot change or correct a peer that is sending an excessive number of prefixes. The network operator will need to reconfigure the maximum-prefix limit or reduce the number of prefixes that are sent from the peer. A peer that is configured to send too many prefixes can cause instability in the network, where an excessive number of prefixes are rapidly advertised and withdrawn. In this case, the **warning-only** keyword of the **neighbor maximum-prefix** command can be configured to disable the restart capability while the network operator corrects the underlying problem.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address***%** | *peer-group-name*} **peer-group** *peer-group-name*
6. **neighbor** {*ip-address* | *ipv6-address***%** | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number*...]
7. **neighbor** {*ip-address* | *ipv6-address***%** | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number*...]
8. **neighbor** {*ip-address* | *ipv6-address***%** | } **maximum-prefix** *maximum* [*threshold*] [**restart** *minutes*] [**warning-only**]
9. **end**
10. **show ip bgp neighbors** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 101`	Enters router configuration mode and creates a BGP routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **peer-group** **Example:** `Device(config-router)# neighbor internal peer-group`	Creates a BGP or multiprotocol BGP peer group.
Step 5	**neighbor** {*ip-address* | *ipv6-address***%** | *peer-group-name*} **peer-group** *peer-group-name* **Example:** `Device(config-router)# neighbor 10.4.9.5 peer-group internal`	Configures a BGP neighbor to member of a peer group. • **%** keyword is the IPv6 link-local address identifier. This keyword needs to be added whenever a link-local IPv6 address is used outside the context of its interface.
Step 6	**neighbor** {*ip-address* | *ipv6-address***%** | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number*...] **Example:** `Device(config-router)# neighbor internal remote-as 100`	Adds a peer group to the BGP or multiprotocol BGP neighbor table.
Step 7	**neighbor** {*ip-address* | *ipv6-address***%** | *peer-group-name*} **remote-as** *autonomous-system-number* [**alternate-as** *autonomous-system-number*...] **Example:** `Device(config-router)# neighbor 10.4.9.5 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table.

	Command or Action	Purpose
Step 8	**neighbor** {*ip-address* | *ipv6-address***%** | } **maximum-prefix** *maximum* [*threshold*] [**restart** *minutes*] [**warning-only**] **Example:** `Device(config-router)# neighbor 10.4.9.5 maximum-prefix 1000 90 restart 60`	Configures the maximum-prefix limit on a router that is running BGP. • Use the **restart** keyword and *minutes* argument to configure the router to automatically reestablish a neighbor session that has been disabled because the maximum-prefix limit has been exceeded. The configurable range of *minutes* is from 1 to 65535 minutes. • Use the **warning-only** keyword to configure the device to disable the restart capability to allow you to adjust a peer that is sending too many prefixes. **Note** If the *minutes* argument is not configured, the disabled session will stay down after the maximum-prefix limit is exceeded. This is the default behavior.
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 10	**show ip bgp neighbors** *ip-address* **Example:** `Device# show ip bgp neighbors 10.4.9.5`	(Optional) Displays information about the TCP and BGP connections to neighbors. • In this example, the output from this command will display the maximum prefix limit for the specified neighbor and the configured restart timer value.

Examples

The following sample output from the **show ip bgp neighbors** command verifies that a device has been configured to automatically reestablish disabled neighbor sessions. The output shows that the maximum prefix limit for neighbor 10.4.9.5 is set to 1000 prefixes, the restart threshold is set to 90 percent, and the restart interval is set at 60 minutes.

```
Device# show ip bgp neighbors 10.4.9.5

BGP neighbor is 10.4.9.5,  remote AS 101, internal link
  BGP version 4, remote router ID 10.4.9.5
  BGP state = Established, up for 2w2d
  Last read 00:00:14, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
```

```
    Opens:                  1          1
    Notifications:          0          0
    Updates:                0          0
    Keepalives:         23095      23095
    Route Refresh:          0          0
    Total:              23096      23096
  Default minimum time between advertisement runs is 5 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor versions 1/0 1/0
  Output queue sizes : 0 self, 0 replicated
  Index 2, Offset 0, Mask 0x4
  Member of update-group 2
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:             0          0
    Prefixes Total:               0          0
    Implicit Withdraw:            0          0
    Explicit Withdraw:            0          0
    Used as bestpath:           n/a          0
    Used as multipath:          n/a          0
                                 Outbound    Inbound
  Local Policy Denied Prefixes:  --------    -------
    Total:                                0          0
!Configured maximum number of prefixes and restart interval information!
  Maximum prefixes allowed 1000
  Threshold for warning message 90%, restart interval 60 min
  Number of NLRIs in the update sent: max 0, min 0
  Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.4.9.21, Local port: 179
Foreign host: 10.4.9.5, Foreign port: 11871
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x5296BD2C):
Timer          Starts    Wakeups            Next
Retrans         23098          0             0x0
TimeWait            0          0             0x0
AckHold         23096      22692             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss: 1900546793  snduna: 1900985663  sndnxt: 1900985663     sndwnd:  14959
irs: 2894590641  rcvnxt: 2895029492  rcvwnd:      14978  delrcvwnd:   1406
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 316 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
Datagrams (max data segment is 1460 bytes):
Rcvd: 46021 (out of order: 0), with data: 23096, total data bytes: 438850
Sent: 46095 (retransmit: 0, fastretransmit: 0), with data: 23097, total data by9
```

Troubleshooting Tips

Use the **clear ip bgp** command to reset a BGP connection using BGP soft reconfiguration. This command can be used to clear stored prefixes to prevent a device that is running BGP from exceeding the maximum-prefix limit.

Display of the following error messages can indicate an underlying problem that is causing the neighbor session to become disabled. You should check the values configured for the **neighbor maximum-prefix**

command and the configuration of any peers that are sending an excessive number of prefixes. The following sample error messages are similar to the error messages that may be displayed:

```
00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Up
00:01:14:%BGP-4-MAXPFX:No. of unicast prefix received from 10.10.10.2 reaches 5, max 6
00:01:14:%BGP-3-MAXPFXEXCEED:No.of unicast prefix received from 10.10.10.2:7 exceed limit6
00:01:14:%BGP-5-ADJCHANGE:neighbor 10.10.10.2 Down - BGP Notification sent
00:01:14:%BGP-3-NOTIFICATION:sent to neighbor 10.10.10.2 3/1 (update malformed) 0 byte
```

The **bgp dampening** command can be used to configure the dampening of a flapping route or interface when a peer is sending too many prefixes and causing network instability. Use this command only when troubleshooting or tuning a device that is sending an excessive number of prefixes. For more details about BGP route dampening, see the “Configuring Advanced BGP Features” module.

Configuration Example for BGP Restart Neighbor Session After Max-Prefix Limit Reached

Example: Configuring a Router to Reestablish a Neighbor Session After the Maximum Prefix Limit Reached

The following example sets the maximum number of prefixes allowed from the neighbor at 192.168.6.6 to 2000 and configures the device to reestablish a peering session after 30 minutes if one has been disabled:

```
Device(config)# router bgp 101
Device(config-router)# neighbor internal peer-group
Device(config-router)# neighbor 10.4.9.5 peer-group internal
Device(config-router)# neighbor internal remote-as 100
Device(config-router)# neighbor 10.4.9.5 remote-as 100
Device(config-router)# neighbor 10.4.9.5 maximum-prefix 2000 90 restart 30
Device(config-router)# end
```

Additional References for BGP Restart Neighbor Session After Max-Prefix Limit Reached

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 4486	*Subcodes for BGP Cease Notification Message*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Maximum-Prefix on IOS XE

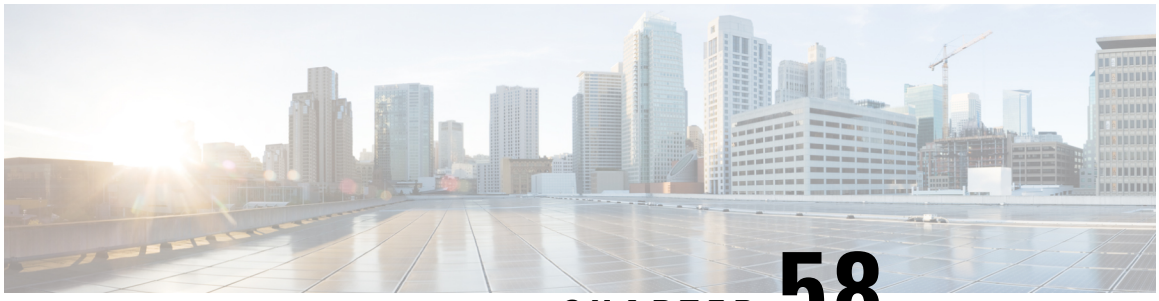
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 83: Feature Information for BGP Maximum-Prefix on IOS XE

Feature Name	Releases	Feature Information
BGP Restart Session After Max-Prefix Limit		The BGP Restart Session After Max-Prefix Limit Reached feature adds the **restart** keyword to the **neighbor maximum-prefix** command. This allows a network operator to configure the time interval at which a peering session is reestablished by a device when the number of prefixes that have been received from a peer has exceeded the maximum prefix limit. The following commands were modified: **neighbor maximum-prefix** and **show ip bgp neighbors**.

Feature Name	Releases	Feature Information
BGP—Subcodes for BGP Cease Notification		Support for subcodes for BGP cease notification has been added.
BGP – Maximum Prefix Discard Extra and Logging enhancement	Cisco IOS XE 17.13.1a	From IOS XE 17.13.1a, BGP Maximum Prefix feature introduces Discard Extra option. This feature drops all excess prefixes received from the neighbor when the configured value of the prefixes exceeds the maximum limit. The Maximum Prefix also introduces a per neighbor enhanced logging time every 60 seconds.



CHAPTER 58

BGP Support for Dual AS Configuration for Network AS Migrations

The BGP Support for Dual AS Configuration for Network AS Migrations feature extended the functionality of the BGP Local-AS feature by providing additional autonomous system path customization configuration options. The configuration of this feature is transparent to customer peering sessions, allowing the provider to merge two autonomous systems without interrupting customer peering arrangements. Customer peering sessions can later be updated during a maintenance window or during other scheduled downtime.

Information About BGP Support for Dual AS Configuration for Network AS Migrations

Autonomous System Migration for BGP Networks

Autonomous system migration can be necessary when a telecommunications or Internet service provider purchases another network. It is desirable for the provider to be able to integrate the second autonomous system without disrupting existing customer peering arrangements. The amount of configuration required in the customer networks can make this a cumbersome task that is difficult to complete without disrupting service.

Dual Autonomous System Support for BGP Network Autonomous System Migration

In Cisco IOS Release 12.0(29)S, 12.3(14)T, 12.2(33)SXH, and later releases, support was added for dual BGP autonomous system configuration to allow a secondary autonomous system to merge under a primary autonomous system, without disrupting customer peering sessions. The configuration of this feature is transparent to customer networks. Dual BGP autonomous system configuration allows a router to appear, to

external peers, as a member of secondary autonomous system during the autonomous system migration. This feature allows the network operator to merge the autonomous systems and then later migrate customers to new configurations during normal service windows without disrupting existing peering arrangements.

The **neighbor local-as** command is used to customize the AS_PATH attribute by adding and removing autonomous system numbers for routes received from eBGP neighbors. This feature allows a router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. This feature simplifies this process of changing the autonomous system number in a BGP network by allowing the network operator to merge a secondary autonomous system into a primary autonomous system and then later update the customer configurations during normal service windows without disrupting existing peering arrangements.

BGP Autonomous System Migration Support for Confederations, Individual Peering Sessions, and Peer Groupings

This feature supports confederations, individual peering sessions, and configurations applied through peer groups and peer templates. If this feature is applied to group peers, the individual peers cannot be customized.

Ingress Filtering During BGP Autonomous System Migration

Autonomous system path customization increases the possibility that routing loops can be created if such customization is misconfigured. The larger the number of customer peerings, the greater the risk. You can minimize this possibility by applying policies on the ingress interfaces to block the autonomous system number that is in transition or routes that have no **local-as** configuration.

Caution BGP prepends the autonomous system number from each BGP network that a route traverses to maintain network reachability information and to prevent routing loops. This feature should be configured only for autonomous system migration and should be deconfigured after the transition has been completed. This procedure should be attempted only by an experienced network operator, as routing loops can be created with improper configuration.

BGP Network Migration to 4-Byte Autonomous System Numbers

The BGP Support for 4-Byte ASN feature introduced support for 4-byte autonomous system numbers. Because of increased demand for autonomous system numbers, in January 2009 the IANA started to allocate 4-byte autonomous system numbers in the range from 65536 to 4294967295.

The Cisco implementation of 4-byte autonomous system numbers supports RFC 4893. RFC 4893 was developed to allow BGP to support a gradual transition from 2-byte autonomous system numbers to 4-byte autonomous system numbers. A new reserved (private) autonomous system number, 23456, was created by RFC 4893 and this number cannot be configured as an autonomous system number in the Cisco IOS CLI.

Migrating your BGP network to 4-byte autonomous system numbers requires some planning. If you are upgrading to an image that supports 4-byte autonomous system numbers, you can still use 2-byte autonomous system numbers. The **show** command output and regular expression match are not changed and remain in asplain (decimal value) format for 2-byte autonomous system numbers regardless of the format configured for 4-byte autonomous system numbers.

To ensure a smooth transition, we recommend that all BGP speakers within an autonomous system that is identified using a 4-byte autonomous system number be upgraded to support 4-byte autonomous system numbers.

For details about steps to perform to upgrade a BGP network to full 4-byte autonomous system support, see the Migration Guide for Explaining 4-Byte Autonomous System white paper.

How to Configure BGP Support for Dual AS Configuration for Network AS Migrations

Configuring Dual AS Peering for Network Migration

Perform this task to configure a BGP peer router to appear to external peers as a member of another autonomous system for the purpose of autonomous system number migration. When the BGP peer is configured with dual autonomous system numbers then the network operator can merge a secondary autonomous system into a primary autonomous system and update the customer configuration during a future service window without disrupting existing peering arrangements.

The **show ip bgp** and **show ip bgp neighbors** commands can be used to verify autonomous system number for entries in the routing table and the status of this feature.

Note

- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for only true eBGP peering sessions. This feature cannot be configured for two peers in different subautonomous systems of a confederation.
- The BGP Support for Dual AS Configuration for Network AS Migrations feature can be configured for individual peering sessions and configurations applied through peer groups and peer templates. If this command is applied to a peer group, the peers cannot be individually customized.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **local-as** [*autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]]
6. **neighbor** *ip-address* **remove-private-as**
7. **end**
8. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter-prefixes** *mask-length*]
9. **show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 40000`	Enters router configuration mode, and creates a BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 45000`	Establishes a peering session with a BGP neighbor.
Step 5	**neighbor** *ip-address* **local-as** [*autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]] **Example:** `Router(config-router)# neighbor 10.0.0.1 local-as 50000 no-prepend replace-as dual-as`	Customizes the AS_PATH attribute for routes received from an eBGP neighbor. • The **replace-as** keyword is used to prepend only the local autonomous system number (as configured with the *ip-address* argument) to the AS_PATH attribute. The autonomous system number from the local BGP routing process is not prepended. • The **dual-as** keyword is used to configure the eBGP neighbor to establish a peering session using the real autonomous-system number (from the local BGP routing process) or by using the autonomous system number configured with the *ip-address* argument (local-as). • The example configures the peering session with the 10.0.0.1 neighbor to accept the real autonomous system number and the local-as number.
Step 6	**neighbor** *ip-address* **remove-private-as** **Example:** `Router(config-router)# neighbor 10.0.0.1 remove-private-as`	(Optional) Removes private autonomous system numbers from outbound routing updates. • This command can be used with the **replace-as** functionality to remove the private autonomous system number and replace it with an external autonomous system number. • Private autonomous system numbers (64512 to 65535) are automatically removed from the AS_PATH attribute when this command is configured.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 8	**show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**] [**prefix-list** *prefix-list-name* | **route-map** *route-map-name*] [**shorter-prefixes** *mask-length*] **Example:** `Router# show ip bgp`	Displays entries in the BGP routing table. • The output can be used to verify if the real autonomous system number or local-as number is configured.
Step 9	**show ip bgp neighbors** [*neighbor-address*] [**received-routes** | **routes** | **advertised-routes** | **paths** *regexp* | **dampened-routes** | **received** *prefix-filter*] **Example:** `Router# show ip bgp neighbors`	Displays information about TCP and BGP connections to neighbors. • The output will display **local AS**, **no-prepend**, **replace-as**, and **dual-as** with the corresponding autonomous system number when these options are configured.

Configuration Examples for Dual-AS Peering for Network Migration

Example: Dual AS Configuration

The following examples shows how this feature is used to merge two autonomous systems without interrupting peering arrangements with the customer network. The **neighbor local-as** command is configured to allow Router 1 to maintain peering sessions through autonomous system 40000 and autonomous system 45000. Router 2 is a customer router that runs a BGP routing process in autonomous system 50000 and is configured to peer with autonomous-system 45000.

Router 1 in Autonomous System 40000 (Provider Network)

```
interface Serial3/0
 ip address 10.3.3.11 255.255.255.0
!
router bgp 40000
 no synchronization
 bgp router-id 10.0.0.11
 neighbor 10.3.3.33 remote-as 50000
 neighbor 10.3.3.33 local-as 45000 no-prepend replace-as dual-as
```

Router 1 in Autonomous System 45000 (Provider Network)

```
interface Serial3/0
 ip address 10.3.3.11 255.255.255.0
```

```
!
router bgp 45000
 bgp router-id 10.0.0.11
 neighbor 10.3.3.33 remote-as 50000
```

Router 2 in Autonomous System 50000 (Customer Network)

```
interface Serial3/0
 ip address 10.3.3.33 255.255.255.0
!
router bgp 50000
 bgp router-id 10.0.0.3
 neighbor 10.3.3.11 remote-as 45000
```

After the transition is complete, the configuration on router 50000 can be updated to peer with autonomous system 40000 during a normal maintenance window or during other scheduled downtime:

```
 neighbor 10.3.3.11 remote-as 100
```

Example: Dual AS Confederation Configuration

The following example can be used in place of the Router 1 configuration in the "Example: Dual AS Configuration" example. The only difference between these configurations is that Router 1 is configured to be part of a confederation.

```
interface Serial3/0/0
 ip address 10.3.3.11 255.255.255.0
!
router bgp 65534
 no synchronization
 bgp confederation identifier 100
 bgp router-id 10.0.0.11
 neighbor 10.3.3.33 remote-as 50000
 neighbor 10.3.3.33 local-as 45000 no-prepend replace-as dual-as
```

Example: Replace an AS with Another AS in Routing Updates

The following example strips private autonomous system 64512 from outbound routing updates for the 10.3.3.33 neighbor and replaces it with autonomous system 50000:

```
router bgp 64512
 neighbor 10.3.3.33 local-as 50000 no-prepend replace-as
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

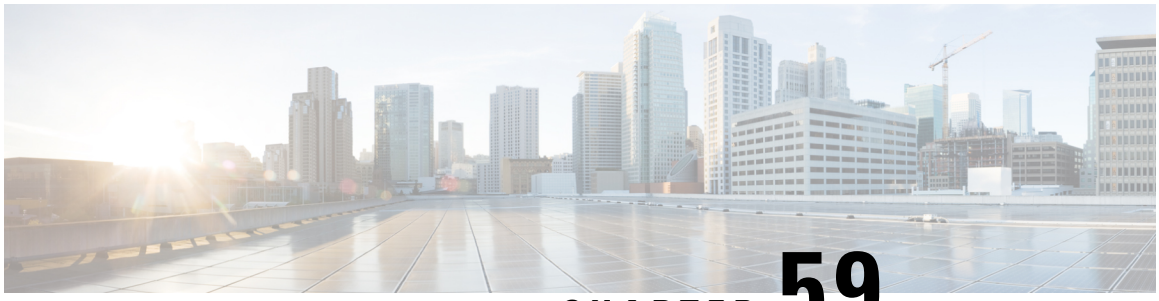
Feature Information for BGP Support for Dual AS Configuration for Network AS Migrations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 84: Feature Information for BGP Support for Dual AS Configuration for Network AS Migrations

Feature Name	Releases	Feature Information
BGP Support for Dual AS Configuration for Network AS Migrations		The BGP Support for Dual AS Configuration for Network AS Migrations feature extended the functionality of the BGP Local-AS feature by providing additional autonomous system path customization configuration options. The configuration of this feature is transparent to customer peering sessions, allowing the provider to merge two autonomous systems without interrupting customer peering arrangements. Customer peering sessions can later be updated during a maintenance window or during other scheduled downtime. The following command was modified by this feature: **neighbor local-as**.



CHAPTER 59

Configuring Internal BGP Features

This module describes how to configure internal Border Gateway Protocol (BGP) features. Internal BGP (iBGP) refers to running BGP on networking devices within one autonomous system. BGP is an interdomain routing protocol designed to provide loop-free routing between separate routing domains (autonomous systems) that contain independent routing policies. Many companies now have large internal networks, and there are many issues involved in scaling the existing internal routing protocols to match the increasing traffic demands while maintaining network efficiency.

Information About Internal BGP Features

BGP Routing Domain Confederation

One way to reduce the internal BGP (iBGP) mesh is to divide an autonomous system into multiple subautonomous systems and group them into a single confederation. To the outside world, the confederation looks like a single autonomous system. Each autonomous system is fully meshed within itself and has a few connections to other autonomous systems in the same confederation. Even though the peers in different autonomous systems have external BGP (eBGP) sessions, they exchange routing information as if they were iBGP peers. Specifically, the next hop, Multi Exit Discriminator (MED) attribute, and local preference information are preserved. This feature allows the you to retain a single Interior Gateway Protocol (IGP) for all of the autonomous systems.

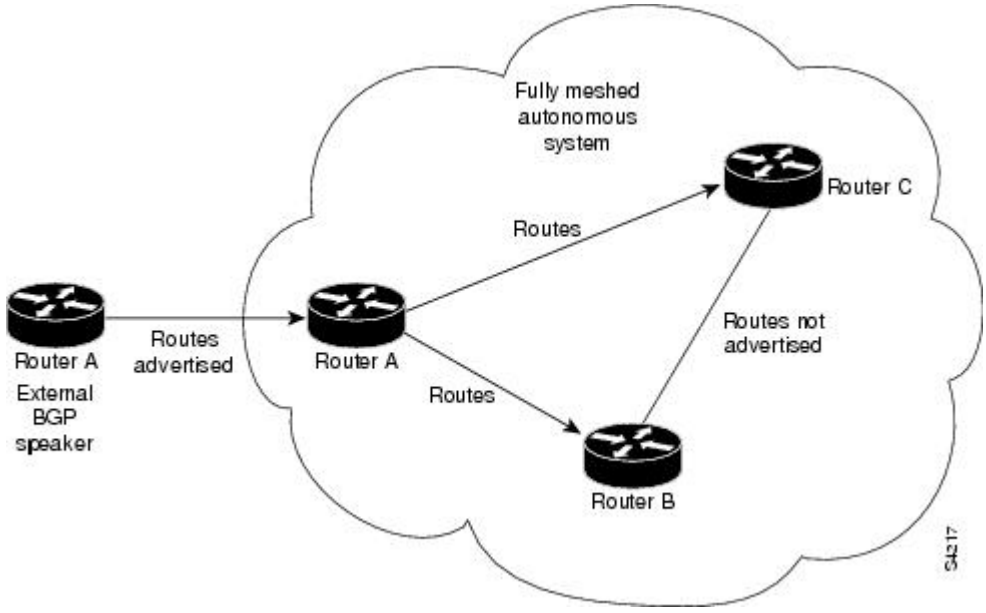
To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number.

BGP Route Reflector

BGP requires that all iBGP speakers be fully meshed. However, this requirement does not scale well when there are many iBGP speakers. Instead of configuring a confederation, another way to reduce the iBGP mesh is to configure a route reflector.

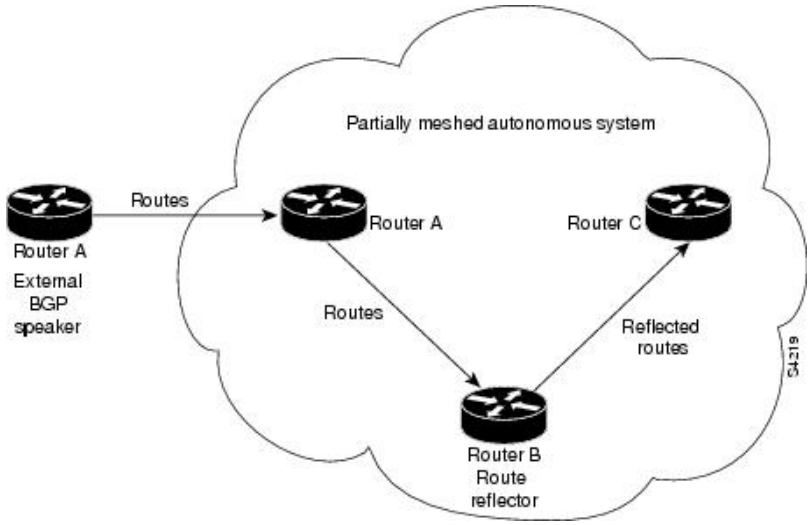
The figure below illustrates a simple iBGP configuration with three iBGP speakers (Routers A, B, and C). Without route reflectors, when Router A receives a route from an external neighbor, it must advertise it to both routers B and C. Routers B and C do not readvertise the iBGP learned route to other iBGP speakers because the routers do not pass on routes learned from internal neighbors to other internal neighbors, thus preventing a routing information loop.

Figure 69: Three Fully Meshed iBGP Speakers



With route reflectors, all iBGP speakers need not be fully meshed because there is a method to pass learned routes to neighbors. In this model, an iBGP peer is configured to be a route reflector responsible for passing iBGP learned routes to a set of iBGP neighbors. In the figure below, Router B is configured as a route reflector. When the route reflector receives routes advertised from Router A, it advertises them to Router C, and vice versa. This scheme eliminates the need for the iBGP session between Routers A and C.

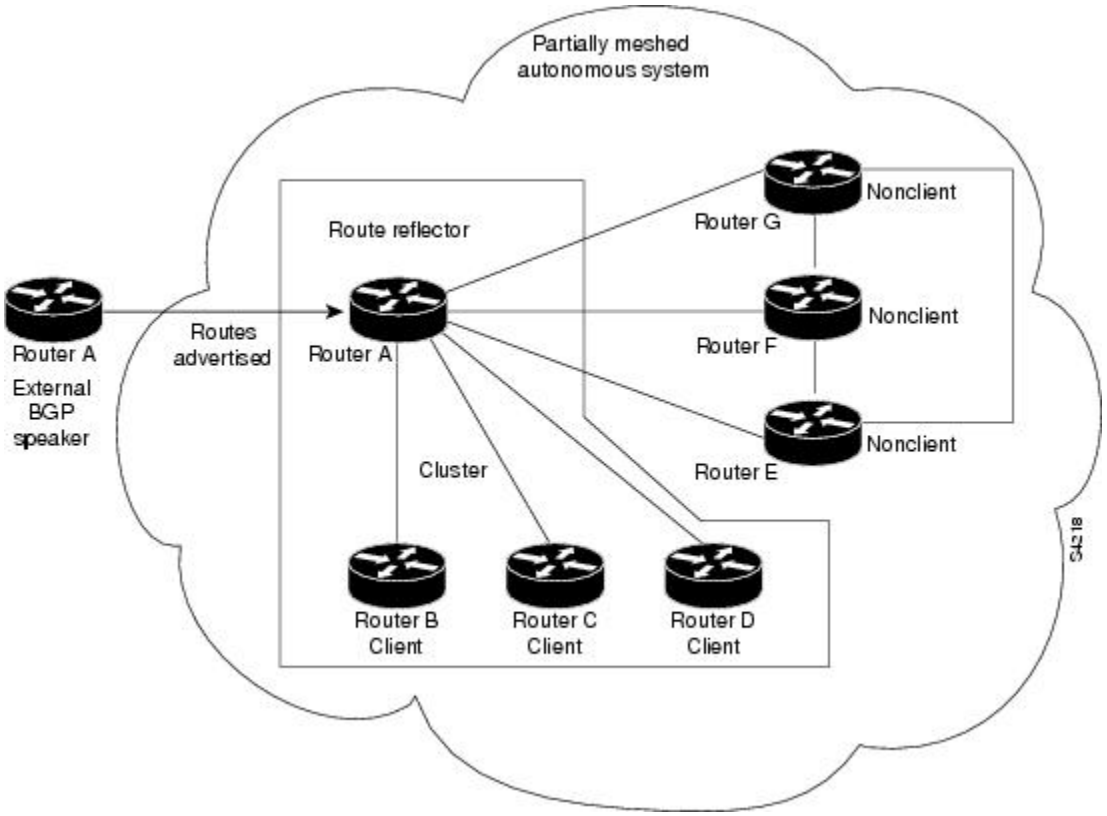
Figure 70: Simple BGP Model with a Route Reflector



The internal peers of the route reflector are divided into two groups: client peers and all the other routers in the autonomous system (nonclient peers). A route reflector reflects routes between these two groups. The route reflector and its client peers form a *cluster*. The nonclient peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with iBGP speakers outside their cluster.

The figure below illustrates a more complex route reflector scheme. Router A is the route reflector in a cluster with routers B, C, and D. Routers E, F, and G are fully meshed, nonclient routers.

Figure 71: More Complex BGP Route Reflector Model



When the route reflector receives an advertised route, depending on the neighbor, it takes the following actions:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.
- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

Along with route reflector-aware BGP speakers, it is possible to have BGP speakers that do not understand the concept of route reflectors. They can be members of either client or nonclient groups allowing an easy and gradual migration from the old BGP model to the route reflector model. Initially, you could create a single cluster with a route reflector and a few clients. All the other iBGP speakers could be nonclient peers to the route reflector and then more clusters could be created gradually.

An autonomous system can have multiple route reflectors. A route reflector treats other route reflectors just like other iBGP speakers. A route reflector can be configured to have other route reflectors in a client group

or nonclient group. In a simple configuration, the backbone could be divided into many clusters. Each route reflector would be configured with other route reflectors as nonclient peers (thus, all the route reflectors will be fully meshed). The clients are configured to maintain iBGP sessions with only the route reflector in their cluster.

Usually a cluster of clients will have a single route reflector. In that case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure, a cluster might have more than one route reflector. In this case, all route reflectors in the cluster must be configured with the 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster. All the route reflectors serving a cluster should be fully meshed and all of them should have identical sets of client and nonclient peers.

Route Reflector Mechanisms to Avoid Routing Loops

As the iBGP learned routes are reflected, routing information may loop. The route reflector model has the following mechanisms to avoid routing loops:

- Originator ID is an optional, nontransitive BGP attribute. It is a 4-byte attribute created by a route reflector. The attribute carries the router ID of the originator of the route in the local autonomous system. Therefore, if a misconfiguration causes routing information to come back to the originator, the information is ignored.
- Cluster-list is an optional, nontransitive BGP attribute. It is a sequence of cluster IDs that the route has passed. When a route reflector reflects a route from its clients to nonclient peers, and vice versa, it appends the local cluster ID to the cluster list. If the cluster list is empty, a new cluster list is created. Using this attribute, a route reflector can identify if routing information is looped back to the same cluster due to misconfiguration. If the local cluster ID is found in the cluster list, the advertisement is ignored.
- The use of **set** clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most **set** clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only **set** clause of an outbound route map that is acted upon is the **set ip next-hop** clause.

BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer

The BGP Outbound Route Map on Route Reflector to Set IP Next Hop feature allows a route reflector to modify the next hop attribute for a reflected route.

The use of **set** clauses in outbound route maps can modify attributes and possibly create routing loops. To avoid this behavior, most **set** clauses of outbound route maps are ignored for routes reflected to iBGP peers. The only **set** clause of an outbound route map on a route reflector (RR) that is acted upon is the **set ip next-hop** clause. The **set ip next-hop** clause is applied to reflected routes.

Configuring an RR with an outbound route map allows a network administrator to modify the next hop attribute for a reflected route. By configuring a route map with the **set ip next-hop** clause, the administrator puts the RR into the forwarding path, and can configure iBGP multipath load sharing to achieve load balancing. That is, the RR can distribute outgoing packets among multiple egress points. See the “Configuring iBGP Multipath Load Sharing” module.

Caution Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should be attempted only by someone who has a good understanding of the design implications.

BGP Route Dampening

Route dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly.

For example, consider a network with three BGP autonomous systems: autonomous system 1, autonomous system 2, and autonomous system 3. Suppose the route to network A in autonomous system 1 flaps (it becomes unavailable). Under circumstances without route dampening, the eBGP neighbor of autonomous system 1 to autonomous system 2 sends a withdraw message to autonomous system 2. The border router in autonomous system 2, in turn, propagates the withdraw message to autonomous system 3. When the route to network A reappears, autonomous system 1 sends an advertisement message to autonomous system 2, which sends it to autonomous system 3. If the route to network A repeatedly becomes unavailable, then available, many withdrawal and advertisement messages are sent. This is a problem in an internetwork connected to the Internet because a route flap in the Internet backbone usually involves many routes.

Note No penalty is applied to a BGP peer reset when route dampening is enabled. Although the reset withdraws the route, no penalty is applied in this instance, even if route flap dampening is enabled.

Route Dampening Minimizes Route Flapping

The route dampening feature minimizes the flapping problem as follows. Suppose again that the route to network A flaps. The router in autonomous system 2 (where route dampening is enabled) assigns network A a penalty of 1000 and moves it to history state. The router in autonomous system 2 continues to advertise the status of the route to neighbors. The penalties are cumulative. When the route flaps so often that the penalty exceeds a configurable suppress limit, the router stops advertising the route to network A, regardless of how many times it flaps. Thus, the route is dampened.

The penalty placed on network A is decayed until the reuse limit is reached, upon which the route is once again advertised. At half of the reuse limit, the dampening information for the route to network A is removed.

BGP Route Dampening Terms

The following terms are used when describing route dampening:

- Flap—A route whose availability alternates repeatedly.
- History state—After a route flaps once, it is assigned a penalty and put into history state, meaning the router does not have the best path, based on historical information.
- Penalty—Each time a route flaps, the router configured for route dampening in another autonomous system assigns the route a penalty of 1000. Penalties are cumulative. The penalty for the route is stored in the BGP routing table until the penalty exceeds the suppress limit. At that point, the route state changes from history to damp.
- Damp state—In this state, the route has flapped so often that the router will not advertise this route to BGP neighbors.
- Suppress limit—A route is suppressed when its penalty exceeds this limit. The default value is 2000.
- Half-life—Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default). The process of reducing the penalty happens every 5 seconds.

- Reuse limit—As the penalty for a flapping route decreases and falls below this reuse limit, the route is unsuppressed. That is, the route is added back to the BGP table and once again used for forwarding. The default reuse limit is 750. The process of unsuppressing routes occurs at 10-second increments. Every 10 seconds, the router finds out which routes are now unsuppressed and advertises them to the world.
- Maximum suppress limit—This value is the maximum amount of time a route can be suppressed. The default value is four times the half-life.

The routes external to an autonomous system learned via iBGP are not dampened. This policy prevent the iBGP peers from having a higher penalty for routes external to the autonomous system.

BGP Route Map Next Hop Self

The BGP Route Map Next Hop Self feature provides a way to override the settings for bgp next-hop unchanged and bgp next-hop unchanged allpath selectively. These settings are global for an address family. For some routes this may not be appropriate. For example, static routes may need to be redistributed with a next hop of self, but connected routes and routes learned via Interior Border Gateway Protocol (IBGP) or Exterior Border Gateway Protocol (EBGP) may continue to be redistributed with an unchanged next hop.

The BGP route map next hop self functionality modifies the existing route map infrastructure to configure a new ip next-hop self setting, which overrides the bgp next-hop unchanged and bgp next-hop unchanged allpaths settings.

The ip next-hop self setting is applicable only to VPNv4 and VPNv6 address families. Routes distributed by protocols other than BGP are not affected.

You configure a new bgp route-map priority setting to inform BGP that the route map will take priority over the settings for bgp next-hop unchanged and bgp next-hop unchanged allpath. The bgp route-map priority setting only impacts BGP. The bgp route-map priority setting has no impact unless you configure the bgp next-hop unchanged or bgp next-hop unchanged allpaths settings.

How to Configure Internal BGP Features

Configuring a Routing Domain Confederation

To configure a BGP confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems will look like a single autonomous system with the confederation identifier as the autonomous system number. To configure a BGP confederation identifier, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# bgp confederation identifier` *`as-number`*	Configures a BGP confederation.

In order to treat the neighbors from other autonomous systems within the confederation as special eBGP peers, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# bgp confederation peers as-number [as-number]`	Specifies the autonomous systems that belong to the confederation.

For an alternative way to reduce the iBGP mesh, see "Configuring a Route Reflector, on page 971."

Configuring a Route Reflector

To configure a route reflector and its clients, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# neighbor {ip-address | peer-group-name} route-reflector-client`	Configures the local router as a BGP route reflector and the specified neighbor as a client.

If the cluster has more than one route reflector, configure the cluster ID by using the following command in router configuration mode:

Command	Purpose
`Router(config-router)# bgp cluster-id cluster-id`	Configures the cluster ID.

Use the **show ip bgp** command to display the originator ID and the cluster-list attributes.

By default, the clients of a route reflector are not required to be fully meshed and the routes from a client are reflected to other clients. However, if the clients are fully meshed, the route reflector need not reflect routes to clients.

To disable client-to-client route reflection, use the **no bgp client-to-client reflection** command in router configuration mode:

Command	Purpose
`Router(config-router)# no bgp client-to-client reflection`	Disables client-to-client route reflection.

Configuring a Route Reflector Using a Route Map to a Set Next Hop for an iBGP Peer

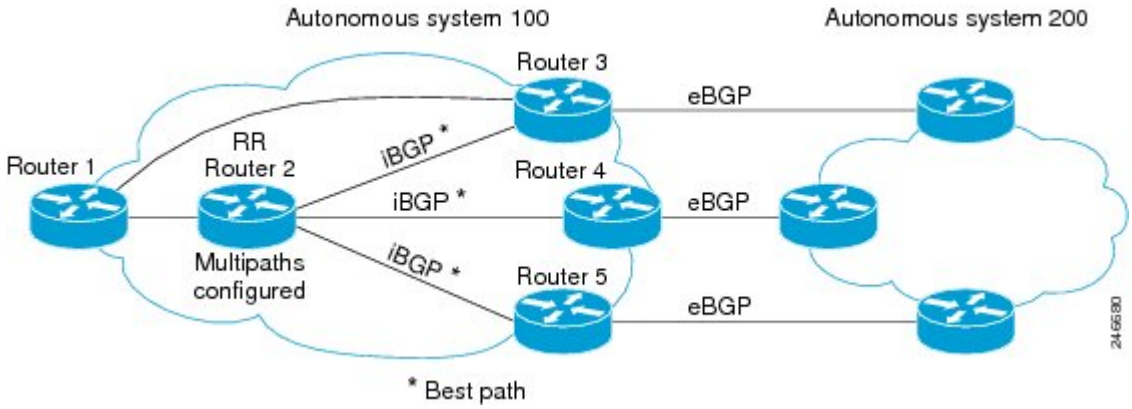
Perform this task on an RR to set a next hop for an iBGP peer. One reason to perform this task is when you want to make the RR the next hop for routes, so that you can configure iBGP load sharing. Create a route map that sets the next hop to be the RR's address, which will be advertised to the RR clients. The route map is applied only to outbound routes from the router to which the route map is applied.

Caution Incorrectly setting BGP attributes for reflected routes can cause inconsistent routing, routing loops, or a loss of connectivity. Setting BGP attributes for reflected routes should only be attempted by someone who has a good understanding of the design implications.

Note Do not use the **neighbor next-hop-self** command to modify the next hop attribute for an RR. Using the **neighbor next-hop-self** command on the RR will modify next hop attributes only for non-reflected routes and not the intended routes that are being reflected from the RR clients. To modify the next hop attribute when reflecting a route, use an outbound route map.

This task configures the RR (Router 2) in the scenario illustrated in the figure below. In this case, Router 1 is the iBGP peer whose routes' next hop is being set. Without a route map, outbound routes from Router 1 would go to next hop Router 3. Instead, setting the next hop to the RR's address will cause routes from Router 1 to go to the RR, and thus allow the RR to perform load balancing among Routers 3, 4, and 5.

Figure 72: Route Reflector Using a Route Map to a Set Next Hop for an iBGP Peer



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag*
4. **set ip next-hop** *ip-address*
5. **exit**
6. **router bgp** *as-number*
7. **address-family ipv4**
8. **maximum-paths ibgp** *number*
9. **neighbor** *ip-address* **remote-as** *as-number*
10. **neighbor** *ip-address* **activate**
11. **neighbor** *ip-address* **route-reflector-client**
12. **neighbor** *ip-address* **route-map** *map-name* **out**
13. Repeat Steps 12 through 14 for the other RR clients.
14. **end**

15. **show ip bgp neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* **Example:** `Router(config)# route-map rr-out`	Enters route map configuration mode to configure a route map. • The route map is created to set the next hop for the route reflector client.
Step 4	**set ip next-hop** *ip-address* **Example:** `Router(config-route-map)# set ip next-hop 10.2.0.1`	Specifies that for routes that are advertised where this route map is applied, the next-hop attribute is set to this IPv4 address. • For this task, we want to set the next hop to be the address of the RR.
Step 5	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 6	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Enters router configuration mode and creates a BGP routing process.
Step 7	**address-family ipv4** **Example:** `Router(config-router-af)# address-family ipv4`	Enters address family configuration mode to configure BGP peers to accept address family specific configurations.
Step 8	**maximum-paths ibgp** *number* **Example:** `Router(config-router)# maximum-paths ibgp 5`	Controls the maximum number of parallel iBGP routes that can be installed in the routing table.

	Command or Action	Purpose
Step 9	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.1.0.1 remote-as 100`	Adds an entry to the BGP neighbor table.
Step 10	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 10.1.0.1 activate`	Enables the exchange of information with the peer.
Step 11	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor 10.1.0.1 route-reflector-client`	Configures the local router as a BGP route reflector, and configures the specified neighbor as a route-reflector client.
Step 12	**neighbor** *ip-address* **route-map** *map-name* **out** **Example:** `Router(config-router-af)# neighbor 10.1.0.1 route-map rr-out out`	Applies the route map to outgoing routes from this neighbor. • Reference the route map you created in Step 3.
Step 13	Repeat Steps 12 through 14 for the other RR clients.	You will not be applying a route map to the other RR clients.
Step 14	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 15	**show ip bgp neighbors** **Example:** `Router# show ip bgp neighbors`	(Optional) Displays information about the BGP neighbors, including their status as RR clients, and information about the route map configured.

Adjusting BGP Timers

BGP uses certain timers to control periodic activities such as the sending of keepalive messages and the interval after not receiving a keepalive message after which the Cisco software declares a peer dead. By default, the keepalive timer is 60 seconds, and the hold-time timer is 180 seconds. You can adjust these timers. When a connection is started, BGP will negotiate the hold time with the neighbor. The smaller of the two hold times will be chosen. The keepalive timer is then set based on the negotiated hold time and the configured keepalive time.

To adjust BGP timers for all neighbors, use the following command in router configuration mode:

Command	Purpose
`Device(config-router)# timers bgp keepalive holdtime`	Adjusts BGP timers for all neighbors.

To adjust BGP keepalive and hold-time timers for a specific neighbor, use the following command in router configuration mode:

Command	Purpose
`Device(config-router)# neighbor [ip-address | peer-group-name] timers keepalive holdtime`	Sets the keepalive and hold-time timers (in seconds) for the specified peer or peer group.

Note The timers configured for a specific neighbor or peer group override the timers configured for all BGP neighbors using the **timers bgp** router configuration command.

To clear the timers for a BGP neighbor or peer group, use the **no** form of the **neighbor timers** command.

Configuring the Router to Consider a Missing MED as the Worst Path

To configure the router to consider a path with a missing MED attribute as the worst path, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# bgp bestpath med missing-as-worst`	Configures the router to consider a missing MED as having a value of infinity, making the path without a MED value the least desirable path.

Configuring the Router to Consider the MED to Choose a Path from Subautonomous System Paths

To configure the router to consider the MED value in choosing a path, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# bgp bestpath med confed`	Configures the router to consider the MED in choosing a path from among those advertised by different subautonomous systems within a confederation.

The comparison between MEDs is made only if there are no external autonomous systems in the path (an external autonomous system is an autonomous system that is not within the confederation). If there is an external autonomous system in the path, then the external MED is passed transparently through the confederation, and the comparison is not made.

The following example compares route A with these paths:

```
path= 65000 65004, med=2
path= 65001 65004, med=3
path= 65002 65004, med=4
path= 65003 1, med=1
```

In this case, path 1 would be chosen if the **bgp bestpath med confed router configuration** command is enabled. The fourth path has a lower MED, but it is not involved in the MED comparison because there is an external autonomous system is in this path.

Configuring the Router to Use the MED to Choose a Path in a Confederation

To configure the router to use the MED to choose the best path from among paths advertised by a single subautonomous system within a confederation, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# bgp deterministic med`	Configures the router to compare the MED variable when choosing among routes advertised by different peers in the same autonomous system.

Note If the **bgp always-compare-med** router configuration command is enabled, all paths are fully comparable, including those from other autonomous systems in the confederation, even if the **bgp deterministic med** command is also enabled.

Enabling and Configuring BGP Route Dampening

Perform this task to enable and configure BGP route dampening.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** as-*number*
4. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
5. **bgp dampening** [*half-life reuse suppress max-suppress-time*] [**route-map** *map-name*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** as-*number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**bgp dampening** [*half-life reuse suppress max-suppress-time*] [**route-map** *map-name*] **Example:** `Router(config-router-af)# bgp dampening 30 1500 10000 120`	Enables BGP route dampening and changes the default values of route dampening factors. • The *half-life*, *reuse*, *suppress*, and *max-suppress-time* arguments are all position dependent; if one argument is entered then all the arguments must be entered. • Use the **route-map** keyword and *map-name* argument to control where BGP route dampening is enabled.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Monitoring and Maintaining BGP Route Dampening

You can monitor the flaps of all the paths that are flapping. The statistics will be deleted once the route is not suppressed and is stable for at least one half-life. To display flap statistics, use the following commands as needed:

Command	Purpose
`Router# show ip bgp dampening flap-statistics`	Displays BGP flap statistics for all paths.
`Router# show ip bgp dampening flap-statistics regexp regexp`	Displays BGP flap statistics for all paths that match the regular expression.
`Router# show ip bgp dampening flap-statistics filter-list access- list`	Displays BGP flap statistics for all paths that pass the filter.
`Router# show ip bgp dampening flap-statistics ip-address mask`	Displays BGP flap statistics for a single entry.
`Router# show ip bgp dampening flap-statistics ip-address mask longer-prefix`	Displays BGP flap statistics for more specific entries.

To clear BGP flap statistics (thus making it less likely that the route will be dampened), use the following commands as needed:

Command	Purpose
`Router# clear ip bgp flap-statistics`	Clears BGP flap statistics for all routes.
`Router# clear ip bgp flap-statistics regexp regexp`	Clears BGP flap statistics for all paths that match the regular expression.
`Router# clear ip bgp flap-statistics filter-list list`	Clears BGP flap statistics for all paths that pass the filter.
`Router# clear ip bgp flap-statistics ip-address mask`	Clears BGP flap statistics for a single entry.
`Router# clear ip bgp ip-address flap-statistics`	Clears BGP flap statistics for all paths from a neighbor.

Note The flap statistics for a route are also cleared when a BGP peer is reset. Although the reset withdraws the route, there is no penalty applied in this instance, even if route flap dampening is enabled.

Once a route is dampened, you can display BGP route dampening information, including the time remaining before the dampened routes will be unsuppressed. To display the information, use the following command:

Command	Purpose
`Router# show ip bgp dampening dampened-paths`	Displays the dampened routes, including the time remaining before they will be unsuppressed.

You can clear BGP route dampening information and unsuppress any suppressed routes by using the following command:

Command	Purpose
`Router# clear ip bgp dampened-paths [ip-address network-mask]`	Clears route dampening information and unsuppresses the suppressed routes.

Configuring BGP Route Map next-hop self

Perform this task to modify the existing route map by adding the ip next-hop self setting and overriding the bgp next-hop unchanged and bgp next-hop unchanged allpaths settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* **permit** *sequence-number*
4. **match source-protocol** *source-protocol*
5. **set ip next-hop self**
6. **exit**
7. **route-map** *map-tag* **permit** *sequence-number*
8. **match route-type internal**
9. **match route-type external**
10. **match source-protocol** *source-protocol*
11. **exit**
12. **router bgp** *autonomous-system-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **next-hop unchanged allpaths**
17. **neighbor** *ip-address* **route-map** *map-name* **out**
18. **exit**
19. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
20. **bgp route-map priority**
21. **redistribute** *protocol*
22. **redistribute** *protocol*
23. **exit-address-family**
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* **permit** *sequence-number* **Example:** `Device(config)# route-map static-nexthop-rewrite permit 10`	Defines conditions for redistributing routes from one routing protocol to another routing protocol and enters route-map configuration mode.
Step 4	**match source-protocol** *source-protocol* **Example:** `Device(config-route-map)# match source-protocol static`	Matches Enhanced Interior Gateway Routing Protocol (EIGRP) external routes based on a source protocol.
Step 5	**set ip next-hop self** **Example:** `Device(config-route-map)# set ip next-hop self`	Configure local routes (for BGP only) with next hop of self.
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 7	**route-map** *map-tag* **permit** *sequence-number* **Example:** `Device(config)# route-map static-nexthop-rewrite permit 20`	Defines conditions for redistributing routes from one routing protocol to another routing protocol and enters route-map configuration mode.
Step 8	**match route-type internal** **Example:** `Device(config-route-map)# match route-type internal`	Redistributes routes of the specified type.
Step 9	**match route-type external** **Example:**	Redistributes routes of the specified type.

	Command or Action	Purpose
	`Device(config-route-map)# match route-type external`	
Step 10	**match source-protocol** *source-protocol* **Example:** `Device(config-route-map)# match source-protocol connected`	Matches Enhanced Interior Gateway Routing Protocol (EIGRP) external routes based on a source protocol.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 12	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 172.16.232.50 remote-as 65001`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 14	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Specifies the VPNv4 address family and enters address family configuration mode.
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 172.16.232.50 activate`	Enables the exchange of information with a Border Gateway Protocol (BGP) neighbor.
Step 16	**neighbor** *ip-address* **next-hop unchanged allpaths** **Example:** `Device(config-router-af)# neighbor 172.16.232.50 next-hop unchanged allpaths`	Enables an external EBGP peer that is configured as multihop to propagate the next hop unchanged.
Step 17	**neighbor** *ip-address* **route-map** *map-name* **out** **Example:** `Device(config-router-af)# neighbor 172.16.232.50 route-map static-nexthop-rewrite out`	Applies a route map to an outgoing route.

	Command or Action	Purpose
Step 18	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.
Step 19	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast vrf inside`	Specifies the IPv4 address family and enters address family configuration mode.
Step 20	**bgp route-map priority** **Example:** `Device(config-router-af)# bgp route-map priority`	Configures the route map priority for the local BGP routing process
Step 21	**redistribute** *protocol* **Example:** `Device(config-router-af)# redistribute static`	Redistributes routes from one routing domain into another routing domain.
Step 22	**redistribute** *protocol* **Example:** `Device(config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain.
Step 23	**exit-address-family** **Example:** `Device(config-router-af)# exit address-family`	Exits address family configuration mode and enters router configuration mode .
Step 24	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Configuration Examples for Internal BGP Features

Example: BGP Confederation Configurations with Route Maps

This section contains an example of the use of a BGP confederation configuration that includes BGP communities and route maps. For more examples of how to configure a BGP confederation, see the “Example: BGP Confederation” section in this module

This example shows how BGP community attributes are used with a BGP confederation configuration to filter routes.

In this example, the route map named *set-community* is applied to the outbound updates to neighbor 172.16.232.50 and the local-as community attribute is used to filter the routes. The routes that pass access list 1 have the special community attribute value local-as. The remaining routes are advertised normally. This special community value automatically prevents the advertisement of those routes by the BGP speakers outside autonomous system 200.

```
router bgp 65000
 network 10.0.1.0 route-map set-community
 bgp confederation identifier 200
 bgp confederation peers 65001
 neighbor 172.16.232.50 remote-as 100
 neighbor 172.16.233.2 remote-as 65001
!
route-map set-community permit 10
 match ip address 1
 set community local-as
!
```

Example: BGP Confederation

The following is a sample configuration that shows several peers in a confederation. The confederation consists of three internal autonomous systems with autonomous system numbers 6001, 6002, and 6003. To the BGP speakers outside the confederation, the confederation looks like a normal autonomous system with autonomous system number 500 (specified via the **bgp confederation identifier** router configuration command).

In a BGP speaker in autonomous system 6001, the **bgp confederation peers** router configuration command marks the peers from autonomous systems 6002 and 6003 as special eBGP peers. Hence peers 172.16.232.55 and 172.16.232.56 will get the local preference, next hop, and MED unmodified in the updates. The router at 10.16.69.1 is a normal eBGP speaker and the updates received by it from this peer will be just like a normal eBGP update from a peer in autonomous system 6001.

```
router bgp 6001
 bgp confederation identifier 500
 bgp confederation peers 6002 6003
 neighbor 172.16.232.55 remote-as 6002
 neighbor 172.16.232.56 remote-as 6003
 neighbor 10.16.69.1 remote-as 777
```

In a BGP speaker in autonomous system 6002, the peers from autonomous systems 6001 and 6003 are configured as special eBGP peers. 10.70.70.1 is a normal iBGP peer and 10.99.99.2 is a normal eBGP peer from autonomous system 700.

```
router bgp 6002
 bgp confederation identifier 500
 bgp confederation peers 6001 6003
 neighbor 10.70.70.1 remote-as 6002
 neighbor 172.16.232.57 remote-as 6001
 neighbor 172.16.232.56 remote-as 6003
 neighbor 10.99.99.2 remote-as 700
```

In a BGP speaker in autonomous system 6003, the peers from autonomous systems 6001 and 6002 are configured as special eBGP peers. 10.200.200.200 is a normal eBGP peer from autonomous system 701.

```
router bgp 6003
 bgp confederation identifier 500
 bgp confederation peers 6001 6002
 neighbor 172.16.232.57 remote-as 6001
 neighbor 172.16.232.55 remote-as 6002
 neighbor 10.200.200.200 remote-as 701
```

The following is a part of the configuration from the BGP speaker 10.200.200.205 from autonomous system 701 in the same example. Neighbor 172.16.232.56 is configured as a normal eBGP speaker from autonomous system 500. The internal division of the autonomous system into multiple autonomous systems is not known to the peers external to the confederation.

```
router bgp 701
 neighbor 172.16.232.56 remote-as 500
 neighbor 10.200.200.205 remote-as 701
```

Example: Route Reflector Using a Route Map to Set a Next Hop for an iBGP Peer

The following example is based on the figure above. Router 2 is the route reflector for the clients: Routers 1, 3, 4, and 5. Router 1 is connected to Router 3, but you don't want Router 1 to forward traffic destined to AS 200 to use Router 3 as the next hop (and therefore use the direct link with Router 3); you want to direct the traffic to the RR, which can load share among Routers 3, 4, and 5.

This example configures the RR, Router 2. A route map named rr-out is applied to Router 1; the route map sets the next hop to be the RR at 10.2.0.1. When Router 1 sees that the next hop is the RR address, Router 1 forwards the routes to the RR. When the RR receives packets, it will automatically load share among the iBGP paths. A maximum of five iBGP paths are allowed.

Router 2

```
route-map rr-out
 set ip next-hop 10.2.0.1
!
interface gigabitethernet 0/0
 ip address 10.2.0.1 255.255.0.0
router bgp 100
 address-family ipv4 unicast
 maximum-paths ibgp 5
 neighbor 10.1.0.1 remote-as 100
 neighbor 10.1.0.1 activate
 neighbor 10.1.0.1 route-reflector-client
 neighbor 10.1.0.1 route-map rr-out out
!
 neighbor 10.3.0.1 remote-as 100
 neighbor 10.3.0.1 activate
 neighbor 10.3.0.1 route-reflector-client
!
 neighbor 10.4.0.1 remote-as 100
 neighbor 10.4.0.1 activate
 neighbor 10.4.0.1 route-reflector-client
!
 neighbor 10.5.0.1 remote-as 100
 neighbor 10.5.0.1 activate
 neighbor 10.5.0.1 route-reflector-client
end
```

Example: Configuring BGP Route Map next-hop self

This section contains an example of how to configure BGP Route Map next-hop self.

In this example, a route map is configured that matches the networks where you wish to override settings for bgp next-hop unchanged and bgp next-hop unchanged allpath. Subsequently, next-hop self is configured. After this, the bgp route map priority is configured for the specified address family so that the previously specified route map takes priority over the settings for bgp next-hop unchanged and bgp next-hop unchanged allpath. This configuration results in static routes being redistributed with a next hop of self, but connected routes and routes learned via IBGP or EBGP continue to be redistributed with an unchanged next hop.

```
route-map static-nexthop-rewrite permit 10
 match source-protocol static
 set ip next-hop self
route-map static-nexthop-rewrite permit 20
 match route-type internal
 match route-type external
 match source-protocol connected
!
router bgp 65000
 neighbor 172.16.232.50 remote-as 65001
 address-family vpnv4
   neighbor 172.16.232.50 activate
   neighbor 172.16.232.50 next-hop unchanged allpaths
   neighbor 172.16.232.50 route-map static-nexthop-rewrite out
 exit-address-family
 address-family ipv4 unicast vrf inside
   bgp route-map priority
   redistribute static
   redistribute connected
 exit-address-family
end
```

Additional References for Internal BGP Features

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP overview	“Cisco BGP Overview” module
Basic BGP configuration tasks	“Configuring a Basic BGP Network” module
iBGP multipath load sharing	“iBGP Multipath Load Sharing” module
Connecting to a service provider	“Connecting to a Service Provider Using External BGP” module

Related Topic	Document Title
Configuring features that apply to multiple IP routing protocols	*IP Routing: Protocol-Independent Configuration Guide*

RFCs

RFC	Title
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2519	*A Framework for Inter-Domain Route Aggregation*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 4271	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4893	*BGP Support for Four-octet AS Number Space*
RFC 5396	*Textual Representation of Autonomous system (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

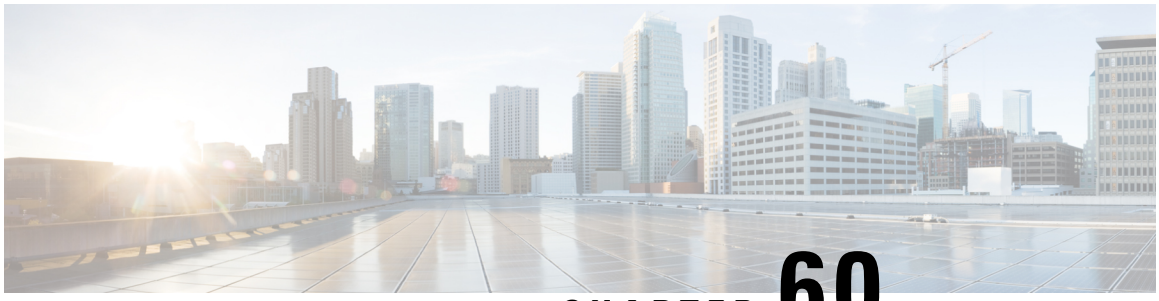
Feature Information for Configuring Internal BGP Features

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 85: Feature Information for Configuring Internal BGP Features

Feature Name	Releases	Feature Configuration Information
Configuring internal BGP features	10.3 12.0(7)T 12.0(32)S12 12.2(33)SRA 12.2(33)SXH	All the features contained in this module are considered to be legacy features and will work in all trains release images. The following commands were introduced or modified by these features: • **bgp always-compare-med** • **bgp bestpath med confed** • **bgp bestpath med missing-as-worst** • **bgp client-to-client reflection** • **bgp cluster-id** • **bgp confederation identifier** • **bgp confederation peers** • **bgp dampening** • **bgp deterministic med** • **clear ip bgp dampening** • **clear ip bgp flap-statistics** • **neighbor route-reflector-client** • **neighbor timers** • **show ip bgp** • **show ip bgp dampening dampened-paths** • **show ip bgp dampening flap-statistics** • **timers bgp**
BGP Outbound Route Map on Route Reflector to Set IP Next Hop	12.0(16)ST 12.0(22)S 12.2 12.2(14)S 15.0(1)S	The BGP Outbound Route Map on Route Reflector to Set IP Next Hop feature allows a route reflector to modify the next hop attribute for a reflected route.



CHAPTER 60

BGP VPLS Auto Discovery Support on Route Reflector

BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector.

Information About BGP VPLS Auto Discovery Support on Route Reflector

BGP VPLS Autodiscovery Support on Route Reflector

In Cisco IOS Release 12.2(33)SRE, BGP VPLS Autodiscovery Support on Route Reflector was introduced. On the Cisco 7600 and Cisco 7200 series routers, BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector. The route reflector reflects the VPLS prefixes to other provider edge (PE) routers so that the PEs do not need to have a full mesh of BGP sessions. The network administrator configures only the BGP VPLS address family on the route reflector.

For an example of a route reflector configuration that can reflect VPLS prefixes, see the “Example: BGP VPLS Autodiscovery Support on Route Reflector” section. For more information about VPLS Autodiscovery, see the “VPLS Autodiscovery BGP Based” module in the *MPLS Layer 2 VPNs Configuration Guide* .

Restrictions for BGP VPLS Auto Discovery Support on Route Reflector

- VPLS BGP Auto Discovery with BGP Signaling in inter-AS Option C is not supported in IOS XE for route reflector.

Configuration Example for BGP VPLS Auto Discovery Support on Route Reflector

Example: BGP VPLS Autodiscovery Support on Route Reflector

In the following example, a host named PE-RR (indicating Provider Edge Route Reflector) is configured as a route reflector capable of reflecting VPLS prefixes. The VPLS address family is configured by **address-family l2vpn vpls** command.

```
hostname PE-RR
!
router bgp 1
 bgp router-id 1.1.1.3
 no bgp default route-target filter
 bgp log-neighbor-changes
neighbor iBGP_PEERS peer-group
neighbor iBGP_PEERS remote-as 1
neighbor iBGP_PEERS update-source Loopback1
neighbor 1.1.1.1 peer-group iBGP_PEERS
neighbor 1.1.1.2 peer-group iBGP_PEERS
!
address-family l2vpn vpls
  neighbor iBGP_PEERS send-community extended
  neighbor iBGP_PEERS route-reflector-client
  neighbor 1.1.1.1 peer-group iBGP_PEERS
  neighbor 1.1.1.2 peer-group iBGP_PEERS
exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

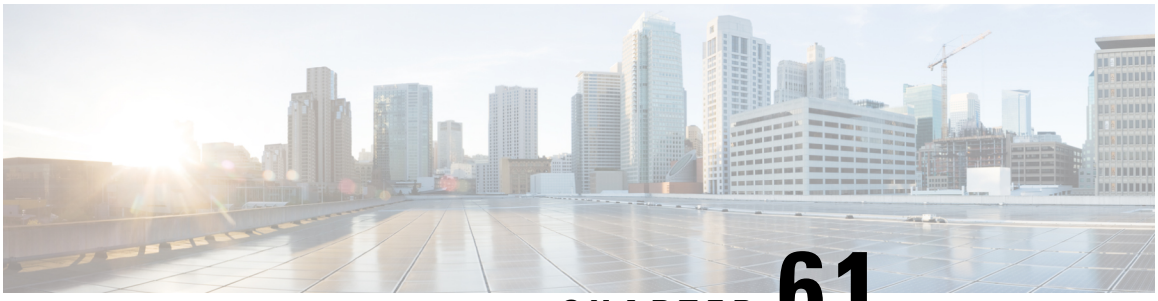
Feature Information for BGP VPLS Auto Discovery Support on Route Reflector

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 86: Feature Information for BGP VPLS Auto Discovery Support on Route Reflector

Feature Name	Releases	Feature Information
BGP VPLS Auto Discovery Support on Route Reflector		BGP Route Reflector was enhanced to be able to reflect BGP VPLS prefixes without having VPLS explicitly configured on the route reflector.



CHAPTER 61

BGP FlowSpec Route-reflector Support

The BGP (Border Gateway Protocol) Flowspec (Flow Specification) Route Reflector feature enables service providers to control traffic flows in their network. This helps in filtering traffic and helps in taking action against distributed denial of service (DDoS) mitigation by dropping the DDoS traffic or diverting it to an analyzer.

BGP flow specification provides a mechanism to encode flow specification rules for traffic flows that can be distributed as BGP Network Layer Reachability Information (NLRI).

Restrictions for BGP FlowSpec Route-reflector Support

- In Cisco IOS 15.5(S) release, BGP flow specification is supported only on a route reflector.
- Mixing of address family matches and actions is not supported in flow spec rules. For example, IPv4 matches cannot be combined with IPv6 actions and vice versa.

Information About BGP FlowSpec Route-reflector Support

Overview of Flowspec

Flowspec specifies procedures for the distribution of flow specification rules as Border Gateway Protocol Network Layer Reachability Information (BGP NLRI) that can be used in any application. It also defines application for the purpose of packet filtering in order to mitigate distributed denial of service attacks.

A flow specification rule consists of a matching part encoded in the BGP NLRI field and an action part encoded as BGP extended community as defined in the RFC 5575. A flow specification rule is a set of data (represented in an n-tuple) consisting of several matching criteria that can be applied to IP packet data. BGP flow specification rules are internally converted to equivalent Cisco Common Classification Policy Language (C3PL) representing corresponding match and action parameters.

In Cisco IOS 15.5(S) release, Flowspec supports following functions for the BGP route reflector:

- Flowspec rules defined in RFC 5575
- IPv6 extensions
- Redirect IP extensions
- BGP flowspec validation

Matching Criteria

The following table lists the various Flowspec tuples that are supported for BGP.

BGP Flowspec NLRI Type	QoS Matching Field (IPv6)	QoS Matching Field (IPv4)	Input Value
Type 1	IPv6 destination address	IPv4 destination address	Prefix length
Type 2	IPv6 source address	IPv4 source address	Prefix length
Type 3	IPv6 next header	IPv4 protocol	Multi-value range
Type 4	IPv6 source or destination port	IPv4 source or destination port	Multi-value range
Type 5	IPv6 destination port	IPv4 destination port	Multi-value range
Type 6	IPv6 source port	IPv4 source port	Multi-value range
Type 7	IPv6 ICMP type	IPv4 ICMP type	Multi-value range
Type 8	IPv6 ICMP code	IPv4 ICMP code	Multi-value range
Type 9	IPv6 TCP flags	IPv4 TCP flags (2 bytes include reserved bits)	Bit mask
Type 10	IPv6 packet length	IPv4 packet length	Multi-value range
Type 11	IPv6 traffic class	IPv4 DSCP	Multi-value range
Type 12	Reserved	IPv4 fragment bits	Bit mask
Type 13	IPv6 flow label	—	Multi-value range

How to Configure BGP FlowSpec Route-reflector Support

Configuring BGP FlowSpec Route-reflector Support

Perform this task to configure BGP FlowSpec on a route reflector. This task specifies only the IPv4 address family but, other address families are also supported for BGP flow specifications.

Before you begin

Configure a BGP route reflector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family {ipv4 | ipv6 | vpnv4 | vpnv6} flowspec**
6. **neighbor** *ip-address* **activate**
7. **neighbor** *ip-address* **route-reflector-client**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 1`	Enters router configuration mode for the BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.1.1.1 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 5	**address-family {ipv4 | ipv6 | vpnv4 | vpnv6} flowspec** **Example:** `Device(config-router)# address-family ipv4 flowspec`	Specifies the address family and enters address family configuration mode. • Flowspec is supported on IPv4, IPv6, VPNv4 and VPNv6 address families.
Step 6	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.1.1.1 activate`	Enables the exchange of information with a BGP neighbor.

	Command or Action	Purpose
Step 7	**neighbor** *ip-address* **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 10.1.1.1 route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client.
Step 8	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits address family configuration mode and returns to privileged EXEC mode.

Disabling BGP FlowSpec Validation

Perform this task if you want to disable the BGP flow specification validations for eBGP peers. The validations are enabled by default.

To know more about BGP flow specification validations, see RFC 5575 (draft-ietf-idr-bgp-flowspec-oid-01-Revised Validation Procedure for BGP Flow Specifications).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family {ipv4 | ipv6 | vpnv4 | vpnv6} flowspec**
5. **neighbor** *ip-address* **validation off**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 1`	Enters router configuration mode for the BGP routing process.
Step 4	**address-family {ipv4 | ipv6 | vpnv4 | vpnv6} flowspec** **Example:** `Device(config-router)# address-family ipv4 flowspec`	Specifies the address family and enters address family configuration mode. • Flowspec is supported on IPv4, IPv6, VPNv4 and VPNv6 address families.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **validation off** **Example:** `Device(config-router-af)# neighbor 10.1.1.1 validation off`	Disables validation of flow specification for eBGP peers.

Verifying BGP FlowSpec Route-reflector Support

The **show** commands can be entered in any order.

Before you begin

Configure BGP FlowSec on a route reflector.

SUMMARY STEPS

1. **show bgp ipv4 flowspec**
2. **show bgp ipv4 flowspec detail**
3. **show bgp ipv4 flowspec summary**
4. **show bgp ipv6 flowspec**
5. **show bgp ipv6 flowspec detail**
6. **show bgp ipv6 flowspec summary**
7. **show bgp vpnv4 flowspec**
8. **show bgp vpnv4 flowspec all detail**
9. **show bgp vpnv6 flowspec**
10. **show bgp vpnv6 flowspec all detail**

DETAILED STEPS

Step 1 show bgp ipv4 flowspec

This command displays the IPv4 flowspec routes.

Example:

```
Device# show bgp ipv4 flowspec

BGP table version is 3, local router ID is 10.10.10.2 Status codes: s suppressed, d damped, h
history,
* valid, > best, i - internal, r RIB-failure, S Stale,
m multipath, b backup-path, f RT-Filter, best-external, a additional-path,
c RIB-compressed, Origin codes: i - IGP, e - EGP, ? - incomplete RPKI validation codes: V valid,
I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i Dest:2.2.2.0/24  10.0.101.1                  100      0 i
 *>i Dest:3.3.3.0/24  10.0.101.1                  100      0 i
```

Step 2 show bgp ipv4 flowspec detail

This command displays the detailed information about IPv4 flowspec routes.

Example:

```
Device# show bgp ipv4 flowspec detail

BGP routing table entry for Dest:2.2.2.0/24, version 2
  Paths: (1 available, best #1, table IPv4-Flowspec-BGP-Table)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local, (Received from a RR-client)
    10.0.101.1 from 10.0.101.1 (10.0.101.1)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: FLOWSPEC Redirect-IP:0x000000000001
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for Dest:3.3.3.0/24, version 3
  Paths: (1 available, best #1, table IPv4-Flowspec-BGP-Table)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local, (Received from a RR-client)
    10.0.101.1 from 10.0.101.1 (10.0.101.1)
      Origin IGP, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

Step 3 **show bgp ipv4 flowspec summary**

This command displays the IPv4 flowspec neighbors.

Example:

```
Device# show bgp ipv4 flowspec summary

BGP router identifier 10.10.10.2, local AS number 239 BGP table version is 3, main routing table
version 3
2 network entries using 16608 bytes of memory
2 path entries using 152 bytes of memory
2/2 BGP path/bestpath attribute entries using 304 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
2 BGP extended community entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory BGP using 17136 total bytes of memory BGP
activity 18/0
prefixes, 18/0 paths, scan interval 15 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
10.0.101.1      4          239      70      24        3    0    0 00:10:58
       2
10.0.101.2      4          239       0       0        1    0    0 never
Idle
10.0.101.3      4          240       0       0        1    0    0 never
Idle
10.10.10.1      4          239      19      23        3    0    0 00:10:53
```

Step 4 **show bgp ipv6 flowspec**

This command displays the IPv6 flowspec routes.

Example:

```
Device# show bgp ipv6 flowspec

BGP table version is 2, local router ID is 10.10.10.2 Status codes: s suppressed, d damped, h
history,
 * valid, > best, i - internal, r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed, Origin codes: i - IGP, e - EGP,
 ? - incomplete RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i Dest:3::/0-24,Source:4::/0-24
                       FEC0::1001                   100      0 i
```

Step 5 **show bgp ipv6 flowspec detail**

This command displays the detailed information about IPv6 flowspec routes.

Example:

```
Device# show bgp ipv6 flowspec detail

BGP routing table entry for Dest:3::/0-24,Source:4::/0-24, version 2
  Paths: (1 available, best #1, table Global-Flowspecv6-Table)
  Advertised to update-groups:
     2
  Refresh Epoch 1
  Local
    FEC0::1001 from FEC0::1001 (10.0.101.2)
      Origin IGP, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

Step 6 **show bgp ipv6 flowspec summary**

This command displays the IPv6 flowspec neighbors.

Example:

```
Device# show bgp ipv6 flowspec summary

BGP router identifier 10.10.10.2, local AS number 239 BGP table version is 3, main routing table
version 3
2 network entries using 16608 bytes of memory
2 path entries using 152 bytes of memory
2/2 BGP path/bestpath attribute entries using 304 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
2 BGP extended community entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory BGP using 17136 total bytes of memory BGP
activity 18/0
prefixes, 18/0 paths, scan interval 15 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
10.0.101.1      4          239      70      24        3    0    0 00:10:58
       2
10.0.101.2      4          239       0       0        1    0    0 never
Idle
10.0.101.3      4          240       0       0        1    0    0 never
Idle
10.10.10.1      4          239      19      23        3    0    0 00:10:53
```

Step 7 **show bgp vpnv4 flowspec**

This command displays the VPNv4 flowspec neighbors.

Example:

```
Device# show bgp vpnv4 flowspec

BGP table version is 2, local router ID is 10.10.10.2 Status codes: s suppressed, d damped, h
history,
* valid, > best, i - internal,r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed, Origin codes: i - IGP, e - EGP,
? - incomplete RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher:  200:200
 *>i Dest:10.0.1.0/24 10.0.101.1                  100      0 i
```

Step 8 **show bgp vpnv4 flowspec all detail**

This command displays the VPNv4 flowspec details.

Example:

```
Device# show bgp vpnv4 flowspec all detail

Route Distinguisher: 200:200
BGP routing table entry for 200:200:Dest:10.0.1.0/24, version 2
  Paths: (1 available, best #1, table VPNv4-Flowspec-BGP-Table)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  Local
    10.0.101.1 (via default) from 10.0.101.1 (10.0.101.1)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:100:100
      rx pathid: 0, tx pathid: 0x0
```

Step 9 **show bgp vpnv6 flowspec**

This command displays the VPNv6 flowspec neighbors.

Example:

```
Device# show bgp vpnv6 flowspec

BGP table version is 2, local router ID is 10.10.10.2 Status codes: s suppressed, d damped, h
history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
             x best-external, a additional-path, c RIB-compressed, Origin codes: i - IGP, e - EGP,
 ? - incomplete RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:200
 *>i SPort:=20640     FEC0::1001                  100      0 i
```

Step 10 **show bgp vpnv6 flowspec all detail**

This command displays the VPNv6 flowspec details.

Example:

```
Device# show bgp vpnv6 flowspec all detail

Route Distinguisher: 200:200
BGP routing table entry for 200:200:SPort:=20640, version 2
```

```
Paths: (1 available, best #1, table VPNv6-Flowspec-BGP-Table)
Advertised to update-groups:
   3
Refresh Epoch 1
Local
  FEC0::1001 (via default) from FEC0::1001 (10.0.101.2)
    Origin IGP, localpref 100, valid, internal, best
    Extended Community: RT:100:100
    rx pathid: 0, tx pathid: 0x0
```

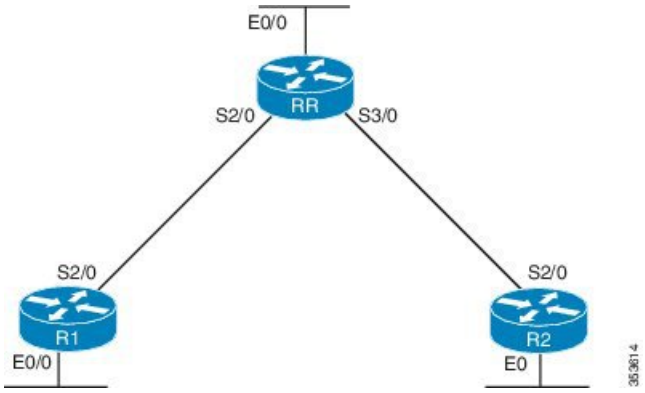
Configuration Examples for BGP FlowSpec Route-reflector Support

Example: BGP FlowSpec Route-reflector Support

Example: Configuring BGP FlowSpec on Route Reflector

Configure BGP route reflector and inject flowspec in the route reflector.

Figure 73: BGP Route Reflector Topology




```
! Configure the topology

!Configure the interfaces on RR

RR> enable
RR# configure terminal
RR(config)# interface E0/0
RR(config-if)# ip address 10.0.0.1 255.224.0.0
RR(config-if)# no shutdown
RR(config-if)# exit
RR(config)# interface S2/0
RR(config-if)# ip address 10.32.0.1 255.224.0.0
RR(config-if)# no shutdown
RR(config-if)# exit
RR(config)# interface S3/0
RR(config-if)# ip address 10.64.0.1 255.224.0.0
```

```
RR(config-if)# no shutdown

!Configure RR as the route reflector with S2/0(R1) and S2/0 (R2) as the neighbors

RR(config)# router bgp 333
RR(config-router)# no synchronization
RR(config-router)# network 10.0.0.0 mask 255.224.0.0
RR(config-router)# network 10.64.0.0 mask 255.224.0.0
RR(config-router)# network 10.32.0.0 mask 255.224.0.0
RR(config-router)# neighbor 10.64.0.2 remote-as 333
RR(config-router)# neighbor 10.32.0.2 remote-as  333


!Configure flowspec on route reflector

RR(config-router)# address-family ipv4 flowspec
RR(configure-router-af)# neighbor 10.64.0.2 activate
RR(config-router)# neighbor 10.64.0.2 route-reflector-client
RR(configure-router-af)# neighbor 10.32.0.2 activate
RR(config-router)# neighbor 10.32.0.2 route-reflector-client

!Verify the configuration

RR> show bgp ipv4 flowspec
```

Additional References for BGP FlowSpec Route-reflector Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5575	*Dissemination of Flow Specification Rules*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

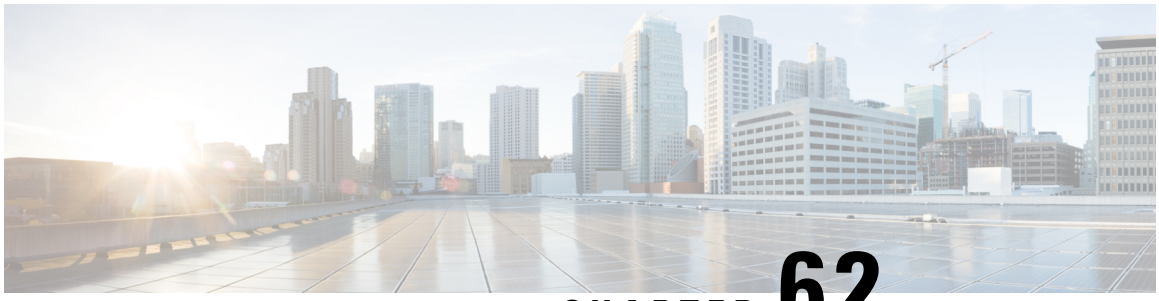
Feature Information for BGP FlowSpec Route-reflector Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 87: Feature Information for BGP FlowSpec Route-reflector Support

Feature Name	Releases	Feature Information
BGP FlowSpec Route-reflector Support	15.5(1)S	The BGP FlowSpec Route-reflector Support feature enables services providers to control traffic flows in their network and mitigate DDoS attack. The following command was introduced by this feature: **address-family {ipv4 | ipv6 | vpnv4 | vpnv6} flowspec**.



CHAPTER 62

BGP Flow Specification Client

The Border Gateway Protocol (BGP) flow specification client feature enables a device to perform the role of a BGP flow specification client and receive flow specification rules from a BGP flow specification controller. Flow specification rules contain a set of match criteria and actions (also called *flows*). The flows are configured on a controller (device), which advertises the flows to the client device, or specific interfaces on the client.

Attention IOS XE software supports BGP flow specification client function and does not support BGP flow specification controller function.

Prerequisites for BGP Flow Specification Client

- Identify and configure flow specification rules on the controller.

Note When the flow specification client is enabled, the matching criteria and corresponding actions in the controller's flows are remotely injected into the client device, and the flows are programmed into the platform hardware of the client device.

Restrictions for BGP Flow Specification Client

- In Cisco IOS 15.5(S) release, BGP flow specification is supported only on a BGP flow specification client and route reflector.

- Mixing of address family matches and actions is not supported in flow specification rules. For example, IPv4 matches cannot be combined with IPv6 actions and vice versa.

Information About BGP Flow Specification Client

BGP Flow Specification Model

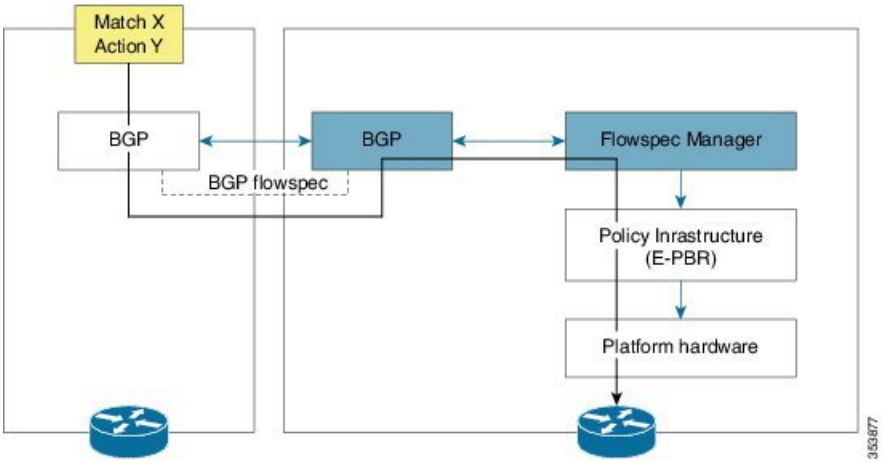
The BGP protocol is used for flow specifications due to unique advantages it offers. The three elements that are used to route flow specifications through BGP enabled devices are: controller, client, and route-reflector (which is optional). This document is specific to the client element function.

Though devices with the IOS XE software (such as ASR 1000, and so on) can perform BGP flow specification client role and not the controller role, a brief outline of the BGP flow specification process is given below for better understanding.

The BGP flow specification functionality allows you to rapidly deploy and propagate filtering and policing functionality among a large number of BGP peer devices to mitigate the effects of a distributed denial-of-service (DDoS) attack over your network.

The BGP flow specification model comprises of a client and a controller (route-reflector usage is optional). The controller is responsible for sending or injecting the flow specification NRLI entry. The client (acting as a BGP speaker) receives the NRLI and programs the hardware forwarding to act on the instruction from the controller. An illustration of this model is provided below.

Figure 74: BGP Flow Specification Model



In the above topology, the controller on the left-hand side injects the flow specification NRLI into the client on the right-hand side. The client receives the information, sends it to the flow specification manager component, configures the ePBR (Enhanced Policy Based Routing) infrastructure, which in turn programs the platform hardware of the device. This way, you can create rules to handle DDoS attacks on your network.

Sample Flow Specification Client Configuration

First, associate the device to a BGP autonomous system and enable flow specification policy mapping capability for various address families. Then, identify a neighbor (through its IP address) as a BGP peer and enable the

capability to exchange information between the devices through the**neighbor activate** command. This way, flow specification information can be exchanged between the client, controller, and any other flow specification client device.

```
!
router bgp 100
 address-family ipv4 flowspec
  neighbor 10.1.1.1 activate
!
```

Matching Criteria and Actions

The flow specification NLRI type consists of several optional sub-components. A specific packet is considered to match the flow specification when it matches the intersection (AND) of all the components present in the specification. The following are the supported component types or tuples that you can define:

BGP Flowspec NLRI Type	QoS Matching Field (IPv6)	QoS Matching Field (IPv4)	Input Value
Type 1	IPv6 destination address	IPv4 destination address	Prefix length
Type 2	IPv6 source address	IPv4 source address	Prefix length
Type 3	IPv6 next header	IPv4 protocol	Multi-value range
Type 4	IPv6 source or destination port	IPv4 source or destination port	Multi-value range
Type 5	IPv6 destination port	IPv4 destination port	Multi-value range
Type 6	IPv6 source port	IPv4 source port	Multi-value range
Type 7	IPv6 ICMP type	IPv4 ICMP type	Multi-value range
Type 8	IPv6 ICMP code	IPv4 ICMP code	Multi-value range
Type 9	IPv6 TCP flags	IPv4 TCP flags (2 bytes include reserved bits)	Bit mask
Type 10	IPv6 packet length	IPv4 packet length	Multi-value range
Type 11	IPv6 traffic class	IPv4 DSCP	Multi-value range
Type 12	Reserved	IPv4 fragment bits	Bit mask

How to Configure BGP Flow Specification Client

Configuring a Device As a Flow Specification Client and Establishing a BGP Peer Relationship With Neighbor

The following task explains configuration of a device as a BGP flow specification client. A device interface within a VRF instance can also perform the role of a BGP flow specification client.

Before you begin

Before configuring a device as a flow specification client, it is a good practice to identify and configure the flow specification controller device (and a route reflector, if required). When flow specification rules are configured on the controller, the rules are remotely injected into the client and the matching criteria and corresponding actions are programmed into the platform hardware of the client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family** {**ipv4** | **ipv6**} **flowspec**
5. **neighbor** *ip-address* **activate**
6. **exit**
7. **address-family** {**ipv4** | **ipv6**} **flowspec vrf** *vrf-name*
8. **neighbor** *ip-address* **remote-as** *as-number*
9. **neighbor** *ip-address* **activate**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Specifies the autonomous system number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 4	**address-family** {**ipv4**	**ipv6**} **flowspec** **Example:**	Specifies either the IPv4 or IPv6 address family and enters BGP address family configuration mode, and initializes

	Command or Action	Purpose
	`Device(config-bgp)# address-family ipv4 flowspec`	the global address family for flow specification policy mapping.
Step 5	**neighbor** *ip-address* **activate** **Example:** `Device(config-bgp-af)# neighbor 10.1.1.1 activate`	Places the device in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer. Enables the device to advertise (and receive information), including its IP address, to its BGP neighbor.
Step 6	**exit** **Example:** `Device(config-bgp-af)# exit`	Exits BGP address family configuration mode and enters BGP configuration mode.
Step 7	**address-family** {**ipv4** | **ipv6**} **flowspec vrf** *vrf-name* **Example:** `Device(config-bgp)# address-family ipv4 flowspec vrf vrf1`	Specifies either the IPv4 or IPv6 address family for the VRF, enters BGP address family configuration mode, and initializes the global address family for flow specification policy mapping.
Step 8	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-bgp-af)# neighbor 2001:DB8:1::1 remote-as 100`	Places the device in neighbor configuration mode for BGP routing and configures the neighbor (IP address) as a BGP peer. The **remote-as** keyword assigns the specified remote autonomous system number to the neighbor.
Step 9	**neighbor** *ip-address* **activate** **Example:** `Device(config-bgp-af)# neighbor 2001:DB8:1::1 activate`	Enables the device to advertise (and receive information), including its IP address, to its BGP neighbor.
Step 10	**exit** **Example:** `Device(config-bgp-af)# exit`	Exits BGP address family configuration mode and enters BGP configuration mode.

Configuring a Flow Specification Policy On All Interfaces Of a Device

The following configuration task explains flow specification policy configuration on all interfaces of a device for the IPv4 and IPv6 address families, and on interfaces within a VRF instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flowspec**
4. **address-family ipv4**
5. **local-install interface-all**
6. **exit**
7. **address-family ipv6**
8. **local-install interface-all**

9. **exit**
10. **vrf** *vrf-name*
11. **address-family ipv4**
12. **local-install interface-all**
13. **exit**
14. **address-family ipv6**
15. **local-install interface-all**
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	**flowspec** **Example:** `Device(config)# flowspec`	Enters flowspec configuration mode.
Step 4	**address-family ipv4** **Example:** `Device(config-flowspec)# address-family ipv4`	Specifies the IPv4 address family and enters flow specification address family configuration mode.
Step 5	**local-install interface-all** **Example:** `Device(config-flowspec-af)# local-install interface-all`	Installs the flowspec policy on all interfaces.
Step 6	**exit** **Example:** `Device(config-flowspec-af)# exit`	Exits flow specification address family configuration mode and enters flowspec configuration mode.
Step 7	**address-family ipv6** **Example:** `Device(config-flowspec)# address-family ipv6`	Specifies the IPv6 address family and enters flow specification address family configuration mode.
Step 8	**local-install interface-all** **Example:** `Device(config-flowspec-af)# local-install interface-all`	Installs the flowspec policy on all interfaces.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-flowspec-af)# exit`	Exits flow specification address family configuration mode and enters flowspec configuration mode.
Step 10	**vrf** *vrf-name* **Example:** `Device(config-flowspec)# vrf vrf10`	Configures a VRF instance and enters flow specification VRF configuration mode.
Step 11	**address-family ipv4** **Example:** `Device(config-flowspec-vrf)# address-family ipv4`	Specifies the IPv4 address family and enters VRF flow specification address family configuration mode.
Step 12	**local-install interface-all** **Example:** `Device(config-flowspec-vrf-af)# local-install interface-all`	Installs the flowspec policy on all interfaces.
Step 13	**exit** **Example:** `Device(config-flowspec-vrf-af)# exit`	Exits VRF flow specification address family configuration mode and enters VRF flow specification configuration mode.
Step 14	**address-family ipv6** **Example:** `Device(config-flowspec-vrf)# address-family ipv6`	Specifies the IPv6 address family and enters VRF flow specification address family configuration mode.
Step 15	**local-install interface-all** **Example:** `Device(config-flowspec-vrf-af)# local-install interface-all`	Installs the flowspec policy on all interfaces.
Step 16	**exit** **Example:** `Device(config-flowspec-vrf-af)# exit`	Exits VRF flow specification address family configuration mode and enters VRF flow specification configuration mode.

Verifying BGP Flow Specification Client

These commands display flow specification configuration details:

SUMMARY STEPS

1. **show flowspec summary**
2. **show bgp ipv4 flowspec**
3. **show flowspec vrf** *vrf-name* **afi-all**

DETAILED STEPS

Step 1 **show flowspec summary**

Example:

```
Device # show flowspec summary

FlowSpec Manager Summary:
Tables: 2
Flows: 1
```

Provides a summary of the flow specification rules present on the node.

In this example, the **Tables** field indicates that the flow specification policy mapping capability is enabled for IPv4 and IPv6 address families.

The **Flows** field indicates that a single flow has been defined across the entire table.

Step 2 **show bgp ipv4 flowspec**

Example:

```
Device # show bgp ipv4 flowspec

Dest:192.0.2.0/24, Source:10.1.1.0/24, DPort:>=120&<=130,SPort:>=25&<=30,DSCP:=30/208
BGP routing table entry for Dest:192.0.2.0/24,
Source:10.1.1.0/24,Proto:=47,DPort:>=120&<=130,SPort:>=25&<=30,DSCP:=30/208 <snip>
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.3
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
0.3 Local
0.0.0.0 from 0.0.0.0 (3.3.3.3)
Origin IGP, localpref 100, valid, redistributed, best, group-best
Received Path ID 0, Local Path ID 1, version 42
Extended community: FLOWSPEC Traffic-rate:100,0
```

Use this command to verify if a flow specification rule configured on the flow specification controller (device) is available on the BGP side. In this example, *redistributed* indicates that the flow specification rule is not internally originated, but one that has been redistributed from the flow specification process to BGP. The extended community (the BGP attribute used to send the match and action criteria to peer devices) that is configured is also displayed.

In this example, the action defined is to rate limit the traffic.

Step 3 **show flowspec vrf** *vrf-name* **afi-all**

Example:

```
Device # show flowspec vrf vrf100 afi-all

VRF: vrf100     AFI: IPv4
    Flow         :DPort:=101,SPort:=101,TCPFlags:~0xFF,Length:>=100&<=1500,DSCP:=63
    Actions      :Redirect: VRF vrf200 Route-target: ASN2-200:2  (bgp.1)
    Flow         :DPort:=102,SPort:=102,TCPFlags:~0xFF,Length:>=100&<=1500,DSCP:=63
    Actions      :Redirect: VRF vrf200 Route-target: ASN2-200:2  (bgp.1)
```

Use this command to verify if a flow specification rule is in a specific VRF associated with the flow specification client (device).

Configuration Examples for BGP Flow Specification Client

Example: Configuring a Device As a Flow Specification Client and Establishing a BGP Peer Relationship With Neighbor

```
Device> enable
Device# configure terminal
Device (config)# router bgp 100
Device (config-bgp)# address-family ipv4 flowspec
Device (config-bgp-af)# neighbor 10.1.1.1 activate
Device (config-bgp-af)# exit
Device (config-bgp)# address-family ipv4 flowspec vrf vrf1
Device (config-bgp-af)# neighbor 2001:DB8:1::1 remote as 100
Device (config-bgp-af)# neighbor 2001:DB8:1::1 activate
Device (config-bgp-af)# exit
```

Example: Configuring a Flow Specification Policy On All Interfaces Of a Device

```
Device> enable
Device# configure terminal
Device(config)# flowspec
Device(config-flowspec)# address-family ipv4
Device(config-flowspec-af)# local-install interface-all
Device(config-flowspec-af)# exit
Device(config-flowspec)# address-family ipv6
Device(config-flowspec-af)# local-install interface-all
Device(config-flowspec-af)#  exit
Device(config-flowspec)# vrf vrf10
Device(config-flowspec-vrf)# address-family ipv4
Device(config-flowspec-vrf-af)# local-install interface-all
Device(config-flowspec-vrf-af)# exit
Device(config-flowspec-vrf)# address-family ipv6
Device(config-flowspec-vrf-af)# local-install interface-all
Device(config-flowspec-vrf-af)# exit
```

Additional References for BGP Flow Specification Client

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP Flow Specification Route-reflector Support	*IP Routing: BGP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 5575	*Dissemination of Flow Specification Rules*

MIBs

MIB	MIBs Link
• CISCO-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

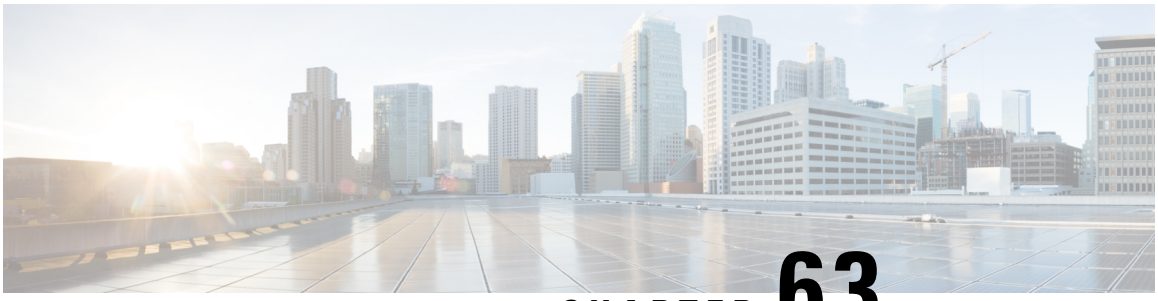
Feature Information for BGP Flow Specification Client

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 88: Feature Information for BGP Flow Specification Client

Feature Name	Releases	Feature Information
BGP Flow Specification Client	Cisco IOS XE 3.15S	The BGP flow specification client feature enables a device to perform the role of a BGP flow specification client and receive flow specification rules from a BGP flow specification controller. The following command was introduced or modified: **flowspec**, **local-install interface-all**.



CHAPTER 63

BGP NSF Awareness

Nonstop Forwarding (NSF) awareness allows a router to assist NSF-capable neighbors to continue forwarding packets during a Stateful Switchover (SSO) operation. The BGP Nonstop Forwarding Awareness feature allows an NSF-aware router that is running BGP to forward packets along routes that are already known for a router that is performing an SSO operation. This capability allows the BGP peers of the failing router to retain the routing information that is advertised by the failing router and continue to use this information until the failed router has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation.

Information About BGP NSF Awareness

Cisco NSF Routing and Forwarding Operation

Cisco NSF is supported by the BGP, EIGRP, OSPF, and IS-IS protocols for routing and by Cisco Express Forwarding (CEF) for forwarding. Of the routing protocols, BGP, EIGRP, OSPF, and IS-IS have been enhanced with NSF capability and awareness, which means that devices running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices.

In this module, a networking device is said to be NSF-aware if it is running NSF-compatible software. A device is said to be NSF-capable if it has been configured to support NSF; therefore, it rebuilds routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. Once the routing protocols have converged, CEF updates the Forwarding Information Base (FIB) table and removes stale route entries. CEF then updates the line cards with the new FIB information.

Cisco Express Forwarding for NSF

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by CEF. CEF maintains the FIB and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active RP synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby RP. Upon switchover of the active RP, the standby RP initially has FIB and adjacency databases that are mirror images of those that were current on the active RP. For platforms with intelligent line cards, the line cards will maintain the current forwarding information over a switchover; for platforms with forwarding engines, CEF will keep the forwarding engine on the standby RP current with changes that are sent to it by CEF on the active RP. In this way, the line cards or forwarding engines will be able to continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates in turn cause prefix-by-prefix updates for CEF, which it uses to update the FIB and adjacency databases. Existing and new entries will receive the new version (epoch) number, indicating that they have been refreshed. The forwarding information is updated on the line cards or forwarding engine during convergence. The RP signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

The routing protocols run only on the active RP, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby RP. After a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables.

Note For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.

BGP Graceful Restart for NSF

When an NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a declaration that the NSF-capable or NSF-aware router has graceful restart capability. Graceful restart is the mechanism by which BGP routing peers avoid a routing flap after a switchover. If the BGP peer has received this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable router and its BGP peer(s) (NSF-aware peers) need to exchange the graceful restart capability in their OPEN messages, at the time of session establishment. If both peers do not exchange the graceful restart capability, the session will not be graceful restart capable.

If the BGP session is lost during the RP switchover, the NSF-aware BGP peer marks all the routes associated with the NSF-capable router as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality means that no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.

After an RP switchover occurs, the NSF-capable router reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. Once this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding

information. The NSF-aware device uses the network information to remove stale routes from its BGP table. Following that, the BGP protocol is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful restart capability in an OPEN message but will establish a BGP session with the NSF-capable device. This functionality will allow interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be graceful restart capable.

BGP NSF Awareness

BGP support for NSF requires that neighbor routers are NSF-aware or NSF-capable. NSF awareness in BGP is also enabled by the graceful restart mechanism. A router that is NSF-aware functions like a router that is NSF-capable with one exception: an NSF-aware router is incapable of performing an SSO operation. However, a router that is NSF-aware is capable of maintaining a peering relationship with an NSF-capable neighbor during an NSF SSO operation, as well as holding routes for this neighbor during the SSO operation.

The BGP Nonstop Forwarding Awareness feature provides an NSF-aware router with the capability to detect a neighbor that is undergoing an SSO operation, maintain the peering session with this neighbor, retain known routes, and continue to forward packets for these routes. The deployment of BGP NSF awareness can minimize the effects of Route Processor (RP) failure conditions and improve the overall network stability by reducing the amount of resources that are normally required for reestablishing peering with a failed router.

NSF awareness for BGP is not enabled by default. The **bgp graceful-restart** command is used to globally enable NSF awareness on a router that is running BGP. NSF-aware operations are also transparent to the network operator and to BGP peers that do not support NSF capabilities.

Note NSF awareness is enabled automatically in supported software images for Interior Gateway Protocols, such as EIGRP, IS-IS, and OSPF. In BGP, global NSF awareness is not enabled automatically and must be started by issuing the **bgp graceful-restart** command in router configuration mode.

How to Configure BGP NSF Awareness

Configuring BGP Nonstop Forwarding Awareness Using BGP Graceful Restart

The tasks in this section show how configure BGP Nonstop Forwarding (NSF) awareness using the BGP graceful restart capability.

- The first task enables BGP NSF globally for all BGP neighbors and suggests a few troubleshooting options.
- The second task describes how to adjust the BGP graceful restart timers, although the default settings are optimal for most network deployments.
- The next three tasks demonstrate how to enable or disable BGP graceful restart for individual BGP neighbors, including peer session templates and peer groups.
- The final task verifies the local and peer router configurations of BGP NSF.

Enabling BGP Global NSF Awareness Using BGP Graceful Restart

Perform this task to enable BGP NSF awareness globally for all BGP neighbors. BGP NSF awareness is part of the graceful restart mechanism and BGP NSF awareness is enabled by issuing the **bgp graceful-restart** command in router configuration mode. BGP NSF awareness allows NSF-aware routers to support NSF-capable routers during an SSO operation. NSF-awareness is not enabled by default and should be configured on all neighbors that participate in BGP NSF.

Note The configuration of the restart and stale-path timers is not required to enable the BGP graceful restart capability. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

Note Configuring both Bidirectional Forwarding Detection (BFD) and BGP graceful restart for NSF on a device running BGP may result in suboptimal routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart** [**restart-time** *seconds*] [**stalepath-time** *seconds*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**bgp graceful-restart** [**restart-time** *seconds*] [**stalepath-time** *seconds*] **Example:**	Enables the BGP graceful restart capability and BGP NSF awareness.

	Command or Action	Purpose
	`Device(config-router)# bgp graceful-restart`	• If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. • Use this command on the restarting router and all of its peers (NSF-capable and NSF-aware).
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

To troubleshoot the NSF feature, use the following commands in privileged EXEC mode, as needed:

- **debug ip bgp** —Displays open messages that advertise the graceful restart capability.
- **debug ip bgp event** —Displays graceful restart timer events, such as the restart timer and the stalepath timer.
- **debug ip bgp updates** —Displays sent and received EOR messages. The EOR message is used by the NSF-aware router to start the stalepath timer, if configured.
- **show ip bgp** —Displays entries in the BGP routing table. The output from this command displays routes that are marked as stale by displaying the letter "S" next to each stale route.
- **show ip bgp neighbor** —Displays information about the TCP and BGP connections to neighbor devices. When enabled, the graceful restart capability is displayed in the output of this command.

What to Do Next

If the **bgp graceful-restart** command has been issued after the BGP session has been established, you must reset by issuing the **clear ip bgp *** command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the **clear ip bgp** command, see the "Configuring a Basic BGP Network" module.

Configuring BGP NSF Awareness Timers

Perform this task to adjust the BGP graceful restart timers. There are two BGP graceful restart timers that can be configured. The optional **restart-time** keyword and *seconds* argument determine how long peer routers will wait to delete stale routes before a BGP open message is received. The default value is 120 seconds. The optional **stalepath-time** keyword and *seconds* argument determine how long a router will wait before deleting stale routes after an end of record (EOR) message is received from the restarting router. The default value is 360 seconds.

Note The configuration of the restart and stale-path timers is not required to enable the BGP graceful restart capability. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart** [**restart-time** *seconds*]
5. **bgp graceful-restart** [**stalepath-time** *seconds*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**bgp graceful-restart** [**restart-time** *seconds*] **Example:** `Device(config-router)# bgp graceful-restart restart-time 130`	Enables the BGP graceful restart capability and BGP NSF awareness. • The **restart-time** argument determines how long peer routers will wait to delete stale routes before a BGP open message is received. • The default value is 120 seconds. The range is from 1 to 3600 seconds. **Note** Only the syntax applicable to this step is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 5	**bgp graceful-restart** [**stalepath-time** *seconds*] **Example:** `Device(config-router)# bgp graceful-restart stalepath-time 350`	Enables the BGP graceful restart capability and BGP NSF awareness. • The **stalepath-time** argument determines how long a router will wait before deleting stale routes after an end of record (EOR) message is received from the restarting router. • The default value is 360 seconds. The range is from 1 to 3600 seconds.

	Command or Action	Purpose
		Note Only the syntax applicable to this step is used in this example. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.

What to Do Next

If the **bgp graceful-restart** command has been issued after the BGP session has been established, you must reset the peer sessions by issuing the **clear ip bgp *** command or by reloading the router before graceful restart capabilities will be exchanged. For more information about resetting BGP sessions and using the **clear ip bgp** command, see the "Configuring a Basic BGP Network" module.

Verifying the Configuration of BGP Nonstop Forwarding Awareness

Use the following steps to verify the local configuration of BGP NSF awareness on a router and to verify the configuration of NSF awareness on peer routers in a BGP network.

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]
3. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config** [*options*]

Displays the running configuration on the local router. The output will display the configuration of the **bgp graceful-restart** command in the BGP section. Repeat this command on all BGP neighbor routers to verify that all BGP peers are configured for BGP NSF awareness. In this example, BGP graceful restart is enabled globally and the external neighbor at 192.168.1.2 is configured to be a BGP peer and will have the BGP graceful restart capability enabled.

Example:

```
Router# show running-config
.
.
.
```

```
router bgp 45000
 bgp router-id 172.17.1.99
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 130
 bgp graceful-restart stalepath-time 350
 bgp graceful-restart
 timers bgp 70 120
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.1.2 activate
.
.
.
```

Step 3 **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

Displays information about TCP and BGP connections to neighbors. “Graceful Restart Capability: advertised” will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. In Cisco IOS Releases 12.2(33)SRC, 12.2(33)SB, or later releases, the ability to enable or disable the BGP graceful restart capability for an individual BGP neighbor, peer group or peer session template was introduced and output was added to this command to show the BGP graceful restart status.

The following partial output example using a Cisco IOS Release 12.2(33)SRC image, displays the graceful restart information for internal BGP neighbor 172.21.1.2 at Router C in the figure above. Note the “Graceful-Restart is enabled” message.

Example:

```
Router# show ip bgp neighbors 172.21.1.2

BGP neighbor is 172.21.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.22.1.1
  BGP state = Established, up for 00:01:01
  Last read 00:00:02, last write 00:00:07, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
!
  Address tracking is enabled, the RIB does have a route to 172.21.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
```

Configuration Examples for BGP NSF Awareness

Example: Enabling BGP Global NSF Awareness Using Graceful Restart

The following example enables BGP NSF awareness globally on all BGP neighbors. The restart time is set to 130 seconds, and the stale path time is set to 350 seconds. The configuration of these timers is optional, and the preconfigured default values are optimal for most network deployments.

```
configure terminal
 router bgp 45000
 bgp graceful-restart
 bgp graceful-restart restart-time 130
 bgp graceful-restart stalepath-time 350
 end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

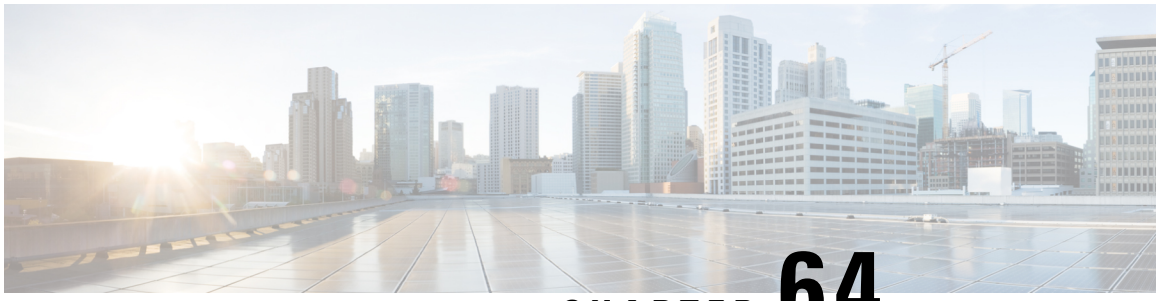
Feature Information for BGP NSF Awareness

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 89: Feature Information for BGP NSF Awareness

Feature Name	Releases	Feature Information
BGP NSF Awareness		Nonstop Forwarding (NSF) awareness allows a device to assist NSF-capable neighbors to continue forwarding packets during a Stateful Switchover (SSO) operation. The BGP Nonstop Forwarding Awareness feature allows an NSF-aware device that is running BGP to forward packets along routes that are already known for a device that is performing an SSO operation. This capability allows the BGP peers of the failing device to retain the routing information that is advertised by the failing device and continue to use this information until the failed device has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation. The following commands were introduced or modified: **bgp graceful-restart**, **show ip bgp**, **show ip bgp neighbors**.



CHAPTER 64

BGP Graceful Restart per Neighbor

The BGP graceful restart feature is already available on a global basis. The BGP Graceful Restart per Neighbor feature allows BGP graceful restart to be enabled or disable for an individual neighbor, providing greater network flexibility and service.

Information About BGP Graceful Restart per Neighbor

BGP Graceful Restart per Neighbor

The ability to enable or disable BGP graceful restart for every individual BGP neighbor was introduced. Three new methods of configuring BGP graceful restart for BGP peers, in addition to the existing global BGP graceful restart configuration, are now available. Graceful restart can be enabled or disabled for a BGP peer or a BGP peer group using the **neighbor ha-mode graceful-restart** command, or a BGP peer can inherit a graceful restart configuration from a BGP peer-session template using the **ha-mode graceful-restart** command.

Although BGP graceful restart is disabled by default, the existing global command enables graceful restart for all BGP neighbors regardless of their capabilities. The ability to enable or disable BGP graceful restart for individual BGP neighbors provides a greater level of control for a network administrator.

When the BGP graceful restart capability is configured for an individual neighbor, each method of configuring graceful restart has the same priority, and the last configuration instance is applied to the neighbor. For example, if global graceful restart is enabled for all BGP neighbors but an individual neighbor is subsequently configured as a member of a peer group for which the graceful restart is disabled, graceful restart is disabled for that neighbor.

The configuration of the restart and stale-path timers is available only with the global **bgp graceful-restart** command, but the default values are set when the **neighbor ha-mode graceful-restart** or **ha-mode graceful-restart** commands are configured. The default values are optimal for most network deployments, and these values should be adjusted only by an experienced network operator.

BGP Peer Session Templates

Peer session templates are used to group and apply the configuration of general BGP session commands to groups of neighbors that share session configuration elements. General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode. Only general session commands can be configured in a peer session template.

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Peer session templates support direct and indirect inheritance. A BGP neighbor can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template. A BGP neighbor can directly inherit only one session template and can indirectly inherit up to seven additional peer session templates.

Peer session templates support inheritance. A directly applied peer session template can directly or indirectly inherit configurations from up to seven peer session templates. So, a total of eight peer session templates can be applied to a neighbor or neighbor group.

Peer session templates support only general session commands. BGP policy configuration commands that are configured only for a specific address family or NLRI configuration mode are configured with peer policy templates.

To use a BGP peer session template to enable or disable BGP graceful restart, see the “Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates” section.

How to Configure BGP Graceful Restart per Neighbor

Enabling BGP Graceful Restart for an Individual BGP Neighbor

Perform this task on Router B in the figure above to enable BGP graceful restart on the internal BGP peer at Router C in the figure above. Under the IPv4 address family, the neighbor at Router C is identified, and BGP graceful restart is enabled for the neighbor at Router C with the IP address 172.21.1.2. To verify that BGP graceful restart is enabled, the optional **show ip bgp neighbors** command is used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **activate**
7. **neighbor** *ip-address* **ha-mode graceful-restart** [**disable**]
8. **end**
9. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 172.21.1.2 remote-as 45000`	Configures peering with a BGP neighbor in the specified autonomous system. • In this example, the BGP peer at 172.21.1.2 is an internal BGP peer because it has the same autonomous system number as the router where the BGP configuration is being entered (see Step 3).
Step 6	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 172.21.1.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local router. • In this example, the internal BGP peer at 172.21.1.2 is activated.
Step 7	**neighbor** *ip-address* **ha-mode graceful-restart** [**disable**] **Example:**	Enables the BGP graceful restart capability for a BGP neighbor. • Use the **disable** keyword to disable BGP graceful restart capability.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 172.21.1.2 ha-mode graceful-restart`	• If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is enabled for the neighbor at 172.21.1.2.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 9	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]] **Example:** `Device# show ip bgp neighbors 172.21.1.2`	(Optional) Displays information about TCP and BGP connections to neighbors. • "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. • In this example, the output is filtered to display information about the BGP peer at 172.21.1.2.

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 172.21.1.2. Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global **bgp graceful-restart** command.

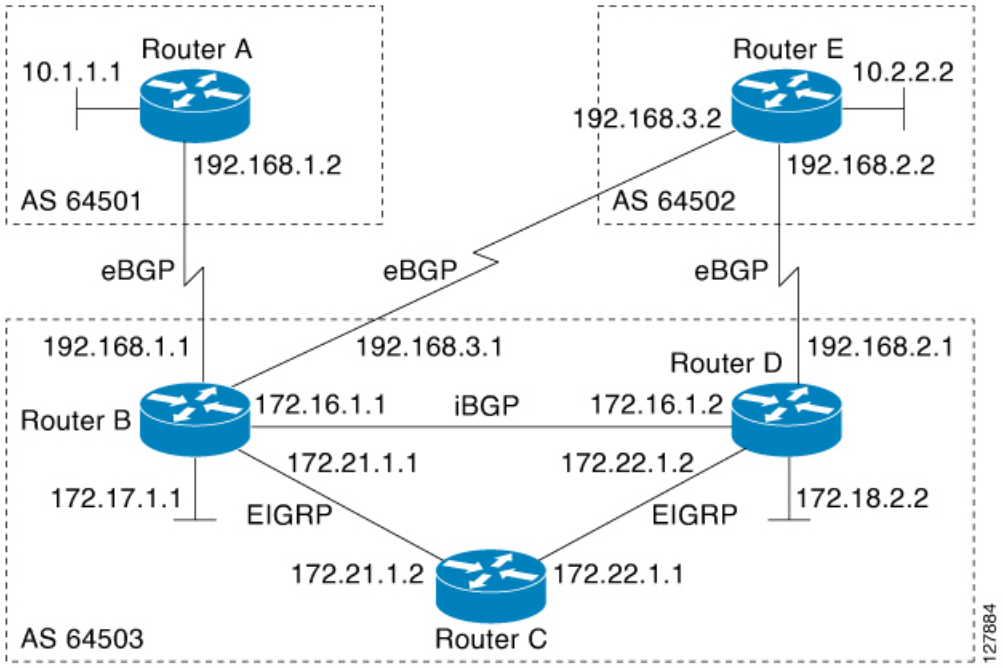
```
Device# show ip bgp neighbors 172.21.1.2

BGP neighbor is 172.21.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 172.22.1.1
  BGP state = Established, up for 00:01:01
  Last read 00:00:02, last write 00:00:07, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
!
  Address tracking is enabled, the RIB does have a route to 172.21.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates

Perform this task to enable and disable BGP graceful restart for BGP neighbors using peer session templates. In this task, a BGP peer session template is created, and BGP graceful restart is enabled. A second peer session template is created, and this template is configured to disable BGP graceful restart.

In this example, the configuration is performed at Router B in the figure below, and two external BGP neighbors—Router A and Router E—are identified. The first BGP peer at Router A is configured to inherit the first peer session template, which enables BGP graceful restart, whereas the second BGP peer at Router E inherits the second template, which disables BGP graceful restart. Using the optional **show ip bgp neighbors** command, the status of the BGP graceful restart capability is verified for each BGP neighbor configured in this task.

Figure 75: Network Topology Showing BGP Neighbors



The restart and stale-path timers can be modified only using the global **bgp graceful-restart** command. The restart and stale-path timers are set to the default values when BGP graceful restart is enabled for BGP neighbors using peer session templates.

Note A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **ha-mode graceful-restart** [**disable**]
6. **exit-peer-session**
7. **template peer-session** *session-template-name*
8. **ha-mode graceful-restart** [**disable**]
9. **exit-peer-session**
10. **bgp log-neighbor-changes**
11. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
12. **neighbor** *ip-address* **inherit peer-session** *session-template-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **neighbor** *ip-address* **inherit peer-session** *session-template-number*
15. **end**
16. **show ip bgp template peer-session** [*session-template-number*]
17. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Device(config-router)# template peer-session S1`	Enters session-template configuration mode and creates a peer session template. • In this example, a peer session template named S1 is created.
Step 5	**ha-mode graceful-restart** [**disable**] **Example:** `Device(config-router-stmp)# ha-mode graceful-restart`	Enables the BGP graceful restart capability and BGP NSF awareness. • Use the **disable** keyword to disable BGP graceful restart capability.

	Command or Action	Purpose
		• If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is enabled for the peer session template named S1.
Step 6	**exit-peer-session** **Example:** `Device(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.
Step 7	**template peer-session** *session-template-name* **Example:** `Device(config-router)# template peer-session S2`	Enters session-template configuration mode and creates a peer session template. • In this example, a peer session template named S2 is created.
Step 8	**ha-mode graceful-restart** [**disable**] **Example:** `Device(config-router-stmp)# ha-mode graceful-restart disable`	Enables the BGP graceful restart capability and BGP NSF awareness. • Use the **disable** keyword to disable BGP graceful restart capability. • If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is disabled for the peer session template named S2.
Step 9	**exit-peer-session** **Example:** `Device(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.
Step 10	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor status changes (up or down) and neighbor resets. • Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 11	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:**	Configures peering with a BGP neighbor in the specified autonomous system.

	Command or Action	Purpose
	`Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	• In this example, the BGP peer at 192.168.1.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).
Step 12	**neighbor** *ip-address* **inherit peer-session** *session-template-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 inherit peer-session S1`	Inherits a peer session template. • In this example, the peer session template named S1 is inherited, and the neighbor inherits the enabling of BGP graceful restart.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.3.2 remote-as 50000`	Configures peering with a BGP neighbor in the specified autonomous system. • In this example, the BGP peer at 192.168.3.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).
Step 14	**neighbor** *ip-address* **inherit peer-session** *session-template-number* **Example:** `Device(config-router)# neighbor 192.168.3.2 inherit peer-session S2`	Inherits a peer session-template. • In this example, the peer session template named S2 is inherited, and the neighbor inherits the disabling of BGP graceful restart.
Step 15	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 16	**show ip bgp template peer-session** [*session-template-number*] **Example:** `Device# show ip bgp template peer-session`	(Optional) Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template by using the *session-template-name* argument. This command also supports all standard output modifiers.
Step 17	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]] **Example:** `Device# show ip bgp neighbors 192.168.1.2`	(Optional) Displays information about TCP and BGP connections to neighbors. • "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. • In this example, the output is filtered to display information about the BGP peer at 192.168.1.2.

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 192.168.1.2 (Router A in the figure above). Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can be set only by using the **bgp graceful-restart** command.

```
Device# show ip bgp neighbors 192.168.1.2

BGP neighbor is 192.168.1.2,  remote AS 40000, external link
 Inherits from template S1 for session parameters
  BGP version 4, remote router ID 192.168.1.2
  BGP state = Established, up for 00:02:11
  Last read 00:00:23, last write 00:00:27, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
!
Address tracking is enabled, the RIB does have a route to 192.168.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 192.168.3.2 (Router E in the figure above). Graceful restart is shown as disabled.

```
Device# show ip bgp neighbors 192.168.3.2

BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Inherits from template S2 for session parameters
  BGP version 4, remote router ID 192.168.3.2
  BGP state = Established, up for 00:01:41
  Last read 00:00:45, last write 00:00:45, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
!
Address tracking is enabled, the RIB does have a route to 192.168.3.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

Disabling BGP Graceful Restart for a BGP Peer Group

Perform this task to disable BGP graceful restart for a BGP peer group. In this task, a BGP peer group is created and graceful restart is disabled for the peer group. A BGP neighbor, Router D at 172.16.1.2 in the

figure above, is then identified and added as a peer group member. It inherits the configuration associated with the peer group, which, in this example, disables BGP graceful restart.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** *peer-group-name* **remote-as** *autonomous-system-number*
7. **neighbor** *peer-group-name* **ha-mode graceful-restart** [**disable**]
8. **neighbor** *ip-address* **peer-group** *peer-group-name*
9. **end**
10. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**unicast**	**multicast**	**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with

	Command or Action	Purpose
		subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router-af)# neighbor PG1 peer-group`	Creates a BGP peer group. • In this example, the peer group named PG1 is created.
Step 6	**neighbor** *peer-group-name* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor PG1 remote-as 45000`	Configures peering with a BGP peer group in the specified autonomous system. • In this example, the BGP peer group named PG1 is added to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 7	**neighbor** *peer-group-name* **ha-mode graceful-restart** [**disable**] **Example:** `Device(config-router-af)# neighbor PG1 ha-mode graceful-restart disable`	Enables the BGP graceful restart capability for a BGP neighbor. • Use the **disable** keyword to disable BGP graceful restart capability. • If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is disabled for the BGP peer group named PG1.
Step 8	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 172.16.1.2 peer-group PG1`	Assigns the IP address of a BGP neighbor to a peer group. • In this example, the BGP neighbor peer at 172.16.1.2 is configured as a member of the peer group named PG1.
Step 9	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 10	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]] **Example:** `Device# show ip bgp neighbors 172.16.1.2`	(Optional) Displays information about TCP and BGP connections to neighbors. • In this example, the output is filtered to display information about the BGP peer at 172.16.1.2 and the "Graceful-Restart is disabled" line shows that the graceful restart capability is disabled for this neighbor.

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 172.16.1.2. Graceful restart is shown as disabled. Note the default values for the restart and stale-path timers. These timers can be set using only the global **bgp graceful-restart** command.

```
Device# show ip bgp neighbors 172.16.1.2

BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
 Member of peer-group PG1 for session parameters
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Neighbor sessions:
    0 active, is multisession capable
!
Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Connections established 0; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
```

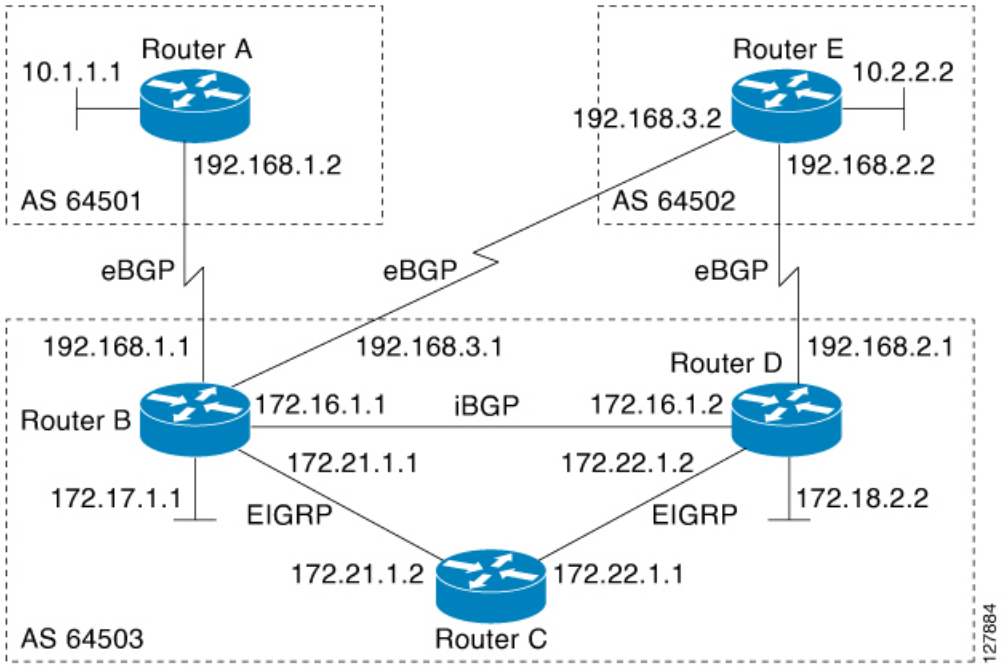
Configuration Examples for BGP Graceful Restart per Neighbor

Enabling and Disabling BGP Graceful Restart Using BGP Peer Session Templates

Perform this task to enable and disable BGP graceful restart for BGP neighbors using peer session templates. In this task, a BGP peer session template is created, and BGP graceful restart is enabled. A second peer session template is created, and this template is configured to disable BGP graceful restart.

In this example, the configuration is performed at Router B in the figure below, and two external BGP neighbors—Router A and Router E—are identified. The first BGP peer at Router A is configured to inherit the first peer session template, which enables BGP graceful restart, whereas the second BGP peer at Router E inherits the second template, which disables BGP graceful restart. Using the optional **show ip bgp neighbors** command, the status of the BGP graceful restart capability is verified for each BGP neighbor configured in this task.

Figure 76: Network Topology Showing BGP Neighbors



The restart and stale-path timers can be modified only using the global **bgp graceful-restart** command. The restart and stale-path timers are set to the default values when BGP graceful restart is enabled for BGP neighbors using peer session templates.

Note A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **ha-mode graceful-restart** [**disable**]
6. **exit-peer-session**
7. **template peer-session** *session-template-name*
8. **ha-mode graceful-restart** [**disable**]
9. **exit-peer-session**
10. **bgp log-neighbor-changes**
11. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
12. **neighbor** *ip-address* **inherit peer-session** *session-template-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **neighbor** *ip-address* **inherit peer-session** *session-template-number*

15. **end**
16. **show ip bgp template peer-session** [*session-template-number*]
17. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Device(config-router)# template peer-session S1`	Enters session-template configuration mode and creates a peer session template. • In this example, a peer session template named S1 is created.
Step 5	**ha-mode graceful-restart** [**disable**] **Example:** `Device(config-router-stmp)# ha-mode graceful-restart`	Enables the BGP graceful restart capability and BGP NSF awareness. • Use the **disable** keyword to disable BGP graceful restart capability. • If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is enabled for the peer session template named S1.
Step 6	**exit-peer-session** **Example:** `Device(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.
Step 7	**template peer-session** *session-template-name* **Example:**	Enters session-template configuration mode and creates a peer session template.

	Command or Action	Purpose
	`Device(config-router)# template peer-session S2`	• In this example, a peer session template named S2 is created.
Step 8	**ha-mode graceful-restart** [**disable**] **Example:** `Device(config-router-stmp)# ha-mode graceful-restart disable`	Enables the BGP graceful restart capability and BGP NSF awareness. • Use the **disable** keyword to disable BGP graceful restart capability. • If you enter this command after the BGP session has been established, you must restart the session in order for the capability to be exchanged with the BGP neighbor. • In this example, the BGP graceful restart capability is disabled for the peer session template named S2.
Step 9	**exit-peer-session** **Example:** `Device(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.
Step 10	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor status changes (up or down) and neighbor resets. • Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 11	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 40000`	Configures peering with a BGP neighbor in the specified autonomous system. • In this example, the BGP peer at 192.168.1.2 is an external BGP peer because it has a different autonomous system number from the router where the BGP configuration is being entered (see Step 3).
Step 12	**neighbor** *ip-address* **inherit peer-session** *session-template-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 inherit peer-session S1`	Inherits a peer session template. • In this example, the peer session template named S1 is inherited, and the neighbor inherits the enabling of BGP graceful restart.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:**	Configures peering with a BGP neighbor in the specified autonomous system. • In this example, the BGP peer at 192.168.3.2 is an external BGP peer because it has a different

	Command or Action	Purpose
	`Device(config-router)# neighbor 192.168.3.2 remote-as 50000`	autonomous system number from the router where the BGP configuration is being entered (see Step 3).
Step 14	**neighbor** *ip-address* **inherit peer-session** *session-template-number* **Example:** `Device(config-router)# neighbor 192.168.3.2 inherit peer-session S2`	Inherits a peer session-template. • In this example, the peer session template named S2 is inherited, and the neighbor inherits the disabling of BGP graceful restart.
Step 15	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and enters privileged EXEC mode.
Step 16	**show ip bgp template peer-session** [*session-template-number*] **Example:** `Device# show ip bgp template peer-session`	(Optional) Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template by using the *session-template-name* argument. This command also supports all standard output modifiers.
Step 17	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]] **Example:** `Device# show ip bgp neighbors 192.168.1.2`	(Optional) Displays information about TCP and BGP connections to neighbors. • "Graceful Restart Capability: advertised" will be displayed for each neighbor that has exchanged graceful restart capabilities with this router. • In this example, the output is filtered to display information about the BGP peer at 192.168.1.2.

Examples

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 192.168.1.2 (Router A in the figure above). Graceful restart is shown as enabled. Note the default values for the restart and stale-path timers. These timers can be set only by using the **bgp graceful-restart** command.

```
Device# show ip bgp neighbors 192.168.1.2

BGP neighbor is 192.168.1.2,  remote AS 40000, external link
 Inherits from template S1 for session parameters
  BGP version 4, remote router ID 192.168.1.2
  BGP state = Established, up for 00:02:11
  Last read 00:00:23, last write 00:00:27, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
```

```
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
    Graceful Restart Capability: advertised
    Multisession Capability: advertised and received
!
Address tracking is enabled, the RIB does have a route to 192.168.1.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is enabled, restart-time 120 seconds, stalepath-time 360 secs
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

The following example shows partial output from the **show ip bgp neighbors** command for the BGP peer at 192.168.3.2 (Router E in the figure above). Graceful restart is shown as disabled.

```
Device# show ip bgp neighbors 192.168.3.2

BGP neighbor is 192.168.3.2,  remote AS 50000, external link
 Inherits from template S2 for session parameters
  BGP version 4, remote router ID 192.168.3.2
  BGP state = Established, up for 00:01:41
  Last read 00:00:45, last write 00:00:45, hold time is 180, keepalive intervals
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
!
Address tracking is enabled, the RIB does have a route to 192.168.3.2
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4724	*Graceful Restart Mechanism for BGP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

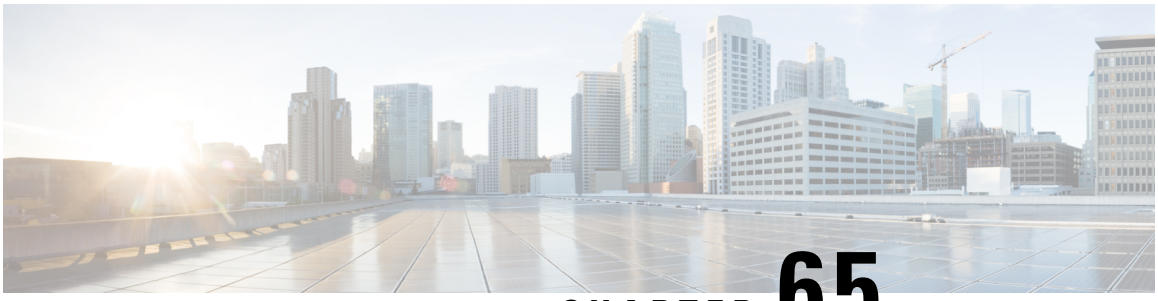
Feature Information for BGP Graceful Restart per Neighbor

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 90: Feature Information for BGP Graceful Restart per Neighbor

Feature Name	Releases	Feature Information
BGP Graceful Restart per Neighbor		The BGP Graceful Restart per Neighbor feature enables or disables the BGP graceful restart capability for an individual BGP neighbor, including using peer session templates and BGP peer groups. The following commands were introduced by this feature: **ha-mode graceful-restart**, and**neighbor ha-mode graceful-restart**. The following command was modified by this feature: **show ip bgp neighbors**.



CHAPTER 65

BGP Support for BFD

Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a significantly faster reconvergence time.

Information About BGP Support for BFD

BFD for BGP

Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a marked decrease in reconvergence time.

See also the “Configuring BGP Neighbor Session Options” chapter, the section “Configuring BFD for BGP IPv6 Neighbors.”

For more details about BFD, see the *Cisco IOS IP Routing: BFD Configuration Guide*.

How to Decrease BGP Convergence Time Using BFD

Prerequisites

- Cisco Express Forwarding (CEF) and IP routing must be enabled on all participating routers.
- BGP must be configured on the routers before BFD is deployed. You should implement fast convergence for the routing protocol that you are using. See the IP routing documentation for your version of Cisco IOS software for information on configuring fast convergence.

Restrictions

- For the Cisco implementation of BFD Support for BGP in Cisco IOS Release15.1(1)SG, only asynchronous mode is supported. In asynchronous mode, either BFD peer can initiate a BFD session.
- IPv6 encapsulation is supported.
- BFD multihop is supported.

Decreasing BGP Convergence Time Using BFD

You start a BFD process by configuring BFD on the interface. When the BFD process is started, no entries are created in the adjacency database, in other words, no BFD control packets are sent or received. The adjacency creation takes places once you have configured BFD support for the applicable routing protocols. The first two tasks must be configured to implement BFD support for BGP to reduce the BGP convergence time. The third task is an optional task to help monitor or troubleshoot BFD.

See also the “Configuring BFD for BGP IPv6 Neighbors” section in the “Configuring BGP Neighbor Session Options” module.

Configuring BFD Session Parameters on the Interface

The steps in this procedure show how to configure BFD on the interface by setting the baseline BFD session parameters on an interface. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 6/0`	Enters interface configuration mode.
Step 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Router(config-if)# bfd interval 50 min_rx 50 multiplier 5`	Enables BFD on the interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode.

Configuring BFD Support for BGP

Perform this task to configure BFD support for BGP, so that BGP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD.

Before you begin

- BGP must be running on all participating routers.
- The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See "Configuring BFD Session Parameters on the Interface" for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **fall-over bfd**

5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp tag1`	Specifies a BGP process and enters router configuration mode.
Step 4	**neighbor** *ip-address* **fall-over bfd** **Example:** `Router(config-router)# neighbor 172.16.10.2 fall-over bfd`	Enables BFD support for fallover.
Step 5	**end** **Example:** `Router(config-router)# end`	Returns the router to privileged EXEC mode.
Step 6	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors detail`	Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	**show ip bgp neighbors** [*ip-address* [**received-routes** | **routes** | **advertised-routes** | **paths** [*regexp*] | **dampened-routes** | **flap-statistics** | **received prefix-filter** | **policy** [**detail**]]] **Example:** `Router# show ip bgp neighbors`	Displays information about BGP and TCP connections to neighbors.

Monitoring and Troubleshooting BFD

To monitor or troubleshoot BFD, perform one or more of the steps in this section.

SUMMARY STEPS

1. **enable**
2. **show bfd neighbors** [**details**]
3. **debug bfd** [**event** | **packet** | **ipc-error** | **ipc-event** | **oir-error** | **oir-event**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show bfd neighbors** [**details**] **Example:** `Router# show bfd neighbors details`	(Optional) Displays the BFD adjacency database. • The **details** keyword shows all BFD protocol parameters and timers per neighbor.
Step 3	**debug bfd** [**event** | **packet** | **ipc-error** | **ipc-event** | **oir-error** | **oir-event**] **Example:** `Router# debug bfd packet`	(Optional) Displays debugging information about BFD packets.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BFD commands	Cisco IOS IP Routing: Protocol Independent Command Reference
Configuring BFD support for another routing protocol	IP Routing: BFD Configuration Guide

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

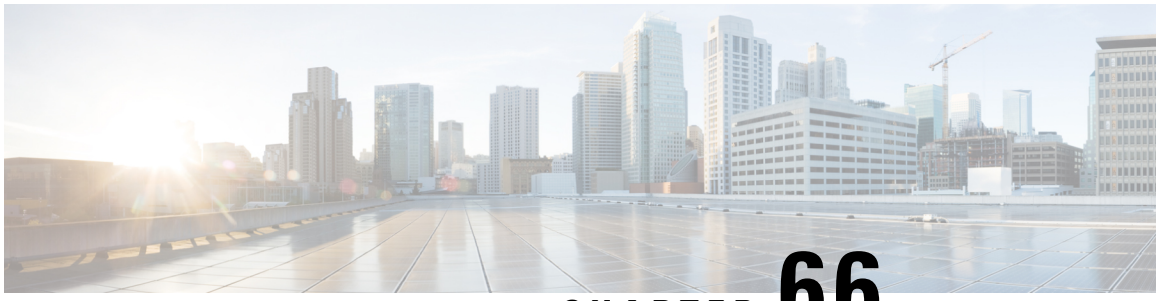
Feature Information for BGP Support for BFD

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 91: Feature Information for BGP Support for BFD

Feature Name	Releases	Feature Information
BGP Support for BFD		Bidirectional Forwarding Detection (BFD) is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable. The main benefit of implementing BFD for BGP is a significantly faster reconvergence time. The following commands were introduced or modified by this feature: **bfd**, **neighbor fall-over**, **show bfd neighbors**, and **show ip bgp neighbors**.



CHAPTER 66

IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

Information About IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

Nonstop Forwarding and Graceful Restart for MP-BGP IPv6 Address Family

The graceful restart capability is supported for IPv6 BGP unicast, multicast, and VPNv6 address families, enabling Cisco nonstop forwarding (NSF) functionality for BGP IPv6. The BGP graceful restart capability allows the BGP routing table to be recovered from peers without keeping the TCP state.

NSF continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. Forwarding is maintained by synchronizing the FIB between the active and standby RP. On switchover, forwarding is maintained using the FIB. The RIB is not kept synchronized; therefore, the RIB is empty on switchover. The RIB is repopulated by the routing protocols and subsequently informs FIB about RIB convergence by using the NSF_RIB_CONVERGED registry call. The FIB tables are updated from the RIB, removing any stale entries. The RIB starts a failsafe timer during RP switchover, in case the routing protocols fail to notify the RIB of convergence.

The Cisco BGP address family identifier (AFI) model is designed to be modular and scalable, and to support multiple AFI and subsequent address family identifier (SAFI) configurations.

How to Configure IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

Configuring the IPv6 BGP Graceful Restart Capability

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
5. **bgp graceful-restart** [**restart-time** *seconds* | **stalepath-time** *seconds*] [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family.
Step 5	**bgp graceful-restart** [**restart-time** *seconds* | **stalepath-time** *seconds*] [**all**] **Example:** `Device(config-router-af)# bgp graceful-restart`	Enables the BGP graceful restart capability.

Configuration Examples for IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

Example: Configuring the IPv6 BGP Graceful Restart Capability

In the following example, the BGP graceful restart capability is enabled:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv6
Device(config-router-af)# bgp graceful-restart
```

In the following example, the restart timer is set to 130 seconds:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv6
Device(config-router-af)# bgp graceful-restart restart-time 130
```

In the following example, the stalepath timer is set to 350 seconds:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv6
Device(config-router-af)# bgp graceful-restart stalepath-time 350
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 NSF and Graceful Restart for MP-BGP IPv6 Address Family

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 92: Feature Information for IPv6: NSF and Graceful Restart for MP-BGP IPv6 Address Family

Feature Name	Releases	Feature Information
IPv6: NSF and Graceful Restart for MP-BGP IPv6 Address Family	Cisco IOS XE Release 3.1	The graceful restart capability is supported for IPv6 BGP unicast, multicast, and VPNv6 address families, enabling Cisco NSF functionality for BGP IPv6. The BGP graceful restart capability allows the BGP routing table to be recovered from peers without keeping the TCP state.

CHAPTER 67

BGP Persistence

BGP persistence enables the router to retain routes that it has learnt from the configured neighbor even when the neighbor session is down. BGP persistence is also referred as long lived graceful restart (LLGR). LLGR comes into effect after graceful restart (GR) ends.

Restrictions for BGP Persistence

- LLGR is not supported for multicast address-families.

Information About BGP Persistence

BGP paths received from neighbor are removed immediately when it detects that the session is down. This behavior is to keep BGP table and forwarding table updated with current network state. Eventually this avoids null routes and routing loops. But in some scenarios keeping the routes for longer time during control plane failure helps the services that are less IP sensitive to continue uninterrupted for longer duration. The traffic flow does not get affected in the following scenarios, even if the BGP routes are stored for longer time during the BGP neighbor failures:

- When route advertisement path is different than the forwarding path that is, through MPLS tunnels. For example, VPN routes.
- When the purpose of route advertisement is to push configuration that is, filter programming on the router. For example, flow-spec, route-targets.
- When route advertisement is used for auto-discovery. For example, VPLS.

BGP persistence enables the local router to retain routes that it has learnt from the configured neighbor even when the neighbor session is down. BGP persistence is also referred as long lived graceful restart (LLGR). LLGR comes into effect after graceful restart (GR) ends. LLGR ends either when the LLGR stale timer expires or when the neighbor sends the end-of-RIB marker after it has sent its routes. When LLGR for a neighbor ends, all routes from that neighbor that are still LLGR stale gets deleted. The LLGR capability is signaled to

a neighbor in the BGP OPEN message, if configured. With BGP persistence the paths are held for very long time (days), and unlike graceful-restart behavior the paths are de-preferenced so that a non-stale path is chosen over a stale path.

BGP speaker advertises LLGR capability including all address-families configured with LLGR. LLGR stale time is per address-family. The BGP persistence feature is supported on the following address family indicators (AFIs):

- VPNv4 and VPNv6
- Flow spec (IPv4, IPv6, VPNv4 and VPNv6)

With BGP persistence, the paths are held for a very long time (days), but unlike basic graceful-restart behavior, the paths are de-preferenced so that a non-stale path is always chosen over a stale path. When the neighbor goes down, it first performs the classic graceful restart which consists of the following steps:

- Starts graceful-restart timer
- Marks the prefixes from its neighbor as stale

Persistence is executed by the helper router only after the graceful-restart is completed. Persistence ends when neighbor sends end-of-row (EoR) or persistence timer expires.

Restart Router

Graceful-restart configuration is mandatory to support persistence (LLGR) neighbor configuration knob to configure persistence. Persistence timer can range between 0 to 4294967.

Helper Router

Helper router executes persistence when graceful-restart is completed. It performs the following tasks:

- Starts persistence timer.
- Marks the prefixes learned from neighbor as **long-lived stale path**.
- Performs best path calculation to de-preference the long lived stale paths. If route has only long lived stale path, it is selected as the best path. If route has multiple long lived stale paths, normal tie-breaking is executed to find the best path.
- Re-advertises the long lived stale path as the best path/add-path with LLGR_STALE (65535:6) community attribute to all LLGR capable configured neighbors.
- Withdraws long lived stale routes from non-LLGR capable neighbors.

Helper Router's Peer

If helper router’s neighbor is LLGR capable, it performs the following for routes received with LLGR_STALE community:

- De-preferences the routes received with LLGR_STALE community attribute.
- Re-advertises the path with LLGR_STALE community attribute as the best path/add-path to the LLGR capable routers with same LLGR_STALE attribute attached.

- Sends route withdraw message to non-LLGR capable routers.

How to Configure BGP Persistence

Configuring BGP Persistence

```
Device# configure terminal
Device(config)# router bgp AS
Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor neighbor -id
Device(config-router-af-nbr)# bgp long-lived-graceful-restart {stale-time send time
 accept time}
```

- **bgp long-lived-graceful-restart**: Enables long lived graceful restart support for the neighbor.
- **stale-time**: Specifies maximum time to wait before purging long-lived stale routes. If the neighbor router negotiates the capability and accept knob is configured locally then lowest of these two values is used as long-lived stale time.
- **send *time***: Specifies stale-time sent in capability. The specified range is between 0-4294967 seconds.
- **accept *time***: Specifies maximum stale-time acceptable from neighbor. The specified range is between 0-4294967 seconds. BGP speaker acts as helper for LLGR capable neighbors. However, it can also act as helper for non-LLGR capable neighbors by configuring accept knob. In that case, value configured with this knob is used as long-lived stale time.

The following is an example:

```
router bgp 1
 address-family vpnv4
  neighbor 1.1.1.1
   long-lived-graceful-restart stale-time send 300 accept 300
   long-lived-graceful-restart stale-time accept 300
```

Verifying BGP Persistence

1. To verify LLGR capability advertise and receive status, use the command **show ip bgp vpnv4 unicast neighbors** *neighbor-id*.

```
show ip bgp vpn4 unicast neighbors 1.1.1.1
……
BGP neighbor is 1.1.1.1,  remote AS 1, internal link
 Description: test1
 …….
Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
    Address family VPNv6 Unicast: advertised and received
    Graceful Restart Capability: advertised and received
```

```
      Remote Restart timer is 10 seconds
      Address families advertised by peer:
        none
      Address families advertised by peer before restart:
        none
    Long-lived Graceful Restart Capability:
        VPNv4 Unicast: advertised and received(was preserved)
        VPNv6 Unicast: received(was preserved)
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1

For address family: VPNv4 Unicast
  Session: 1.1.1.1
……
……
Long-lived Graceful-Restart(was preserved)
    Stalepath-time: sent 2000s, received 50s, accepted 2000s, used 50s
```

2. To verify restarting of the peer, use the command **show ip bgp vpnv4 unicast neighbors** *neighbor-id*.

```
show ip bgp vpn4 unicast neighbors 1.1.1.1
..............
 BGP neighbor is 1.1.1.1,  remote AS 1, internal link
 Description: test1
..............
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Active, down for 00:00:23
  Configured hold time is 15, keepalive interval is 5 seconds
  Minimum holdtime from neighbor is 0 seconds
  Neighbor sessions:
    0 active, is not multisession capable (disabled)
    Stateful switchover support enabled: NO for session 0
  Message statistics:
    InQ depth is 0
…….
…….
For address family: VPNv4 Unicast
  Session: 1.1.1.1
……
……
Long-lived Graceful-Restart
    Stalepath-time: sent 2000s, accepted 2000s, used 2000s
    Stalepath-timer running 37s remaining
```

3. To verify routes marked with long-lived stale routes, use the command **show ip bgp vpnv4 all** .

```
Device# show ip bgp vpnv4 all
BGP table version is 33, local router ID is 19.19.19.19
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
        L long-lived-stale-path
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2:2 (default for vrf example)
 *L>i 20.0.0.0/16      1.1.88.1                 0    100      0 81 ?
 *Li  38.1.1.0/24      1.1.88.1                 0    100      0 81 ?
 *L>i 180.180.180.180/32
                       1.1.88.1                 0    100      0 81 ?
```

```
Router#show ip bgp vpnv4 all 20.0.0.0/16
BGP routing table entry for 2:2:20.0.0.0/16, version 9
Paths: (1 available, best #1, table example)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  81 (long-lived-stale), imported path from 5:5:20.0.0.0/16 (global)
    1.1.88.1 (metric 40) (via default) from 1.1.1.188 (1.1.1.188)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Community: 100:100
      Extended Community: RT:1:1
      Originator: 1.1.88.1, Cluster list: 1.1.1.188
      mpls labels in/out nolabel/44
      rx pathid: 0, tx pathid: 0x0
```

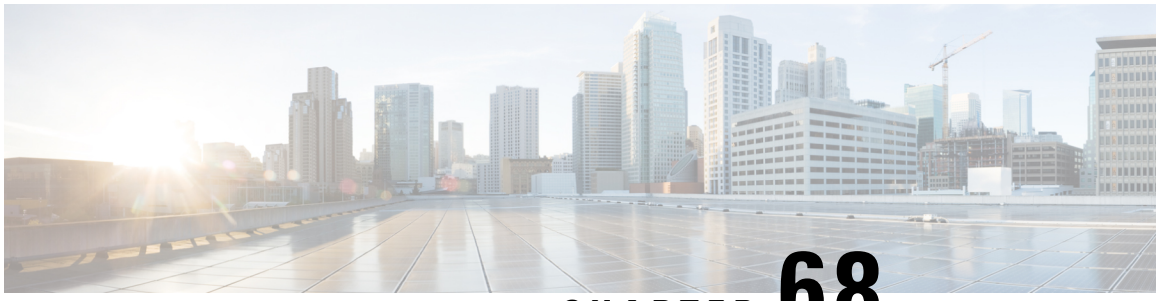
Feature Information for BGP Persistence

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 93: Feature Information for BGP Persistence

Feature Name	Releases	Feature Configuration Information
BGP Persistence	Cisco IOS XE Fuji 16.7.1	BGP persistence enables the router to retain routes that it has learnt from the configured neighbor even after the neighbor session is down. BGP persistence is also referred as Long Lived Graceful Restart (LLGR). The following commands were modified: **address-family vpnv4**, **bgp long-lived-graceful-restart** {*stale-time send time accept time*}, **neighbor** *neighbor-id*, **router bgp** *AS*, **show ip bgp vpnv4 all**, and **show ip bgp vpnv4 unicast neighbors** <*neighbor-id*>.



CHAPTER 68

BGP Link Bandwidth

The Border Gateway Protocol (BGP) Link Bandwidth feature is used to advertise the bandwidth of an autonomous system exit link as an extended community. This feature is configured for links between directly connected external BGP (eBGP) neighbors. The link bandwidth extended community attribute is propagated to iBGP peers when extended community exchange is enabled. This feature is used with BGP multipath features to configure load balancing over links with unequal bandwidth.

Prerequisites for BGP Link Bandwidth

- BGP load balancing or multipath load balancing must be configured before BGP Link Bandwidth feature is enabled.
- BGP extended community exchange must be enabled between iBGP neighbors to which the link bandwidth attribute is to be advertised.
- Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.

Restrictions for BGP Link Bandwidth

- The BGP Link Bandwidth feature can be configured only under IPv4 and VPNv4 address family sessions.
- BGP can originate the link bandwidth community only for directly connected links to eBGP neighbors.
- Both iBGP and eBGP load balancing are supported in IPv4 and VPNv4 address families. However, eiBGP load balancing is supported only in VPNv4 address families.

Information About BGP Link Bandwidth

BGP Link Bandwidth Overview

The BGP Link Bandwidth feature is used to enable multipath load balancing for external links with unequal bandwidth capacity. This feature is enabled under an IPv4 or VPNv4 address family session by entering the **bgp dmzlink-bw** command. This feature supports iBGP, eBGP multipath load balancing, and eiBGP multipath load balancing in Multiprotocol Label Switching (MPLS) VPNs. When this feature is enabled, routes learned from directly connected external neighbor are propagated through the internal BGP (iBGP) network with the bandwidth of the source external link.

The link bandwidth extended community indicates the preference of an autonomous system exit link in terms of bandwidth. This extended community is applied to external links between directly connected eBGP peers by entering the **neighbor dmzlink-bw** command. The link bandwidth extended community attribute is propagated to iBGP peers when extended community exchange is enabled with the **neighbor send-community** command.

Link Bandwidth Extended Community Attribute

The link bandwidth extended community attribute is a 4-byte value that is configured for a link on the demilitarized zone (DMZ) interface that connects two single hop eBGP peers. The link bandwidth extended community attribute is used as a traffic sharing value relative to other paths while traffic is being forwarded. Two paths are designated as equal for load balancing if the weight, local-pref, as-path length, Multi Exit Discriminator (MED), and Interior Gateway Protocol (IGP) costs are the same.

Benefits of the BGP Link Bandwidth Feature

The BGP Link Bandwidth feature allows BGP to be configured to send traffic over multiple iBGP or eBGP learned paths where the traffic that is sent is proportional to the bandwidth of the links that are used to exit the autonomous system. The configuration of this feature can be used with eBGP and iBGP multipath features to enable unequal cost load balancing over multiple links. Unequal cost load balancing over links with unequal bandwidth was not possible in BGP before the BGP Link Bandwidth feature was introduced.

How to Configure BGP Link Bandwidth

Configuring BGP Link Bandwidth

To configure the BGP Link Bandwidth feature, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]

5. **bgp dmzlink-bw**
6. **neighbor** *ip-address* **dmzlink-bw**
7. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 50000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode. • The BGP Link Bandwidth feature is supported only under the IPv4 and VPNv4 address families.
Step 5	**bgp dmzlink-bw** **Example:** `Router(config-router-af)# bgp dmzlink-bw`	Configures BGP to distribute traffic proportionally to the bandwidth of the link. • This command must be entered on each router that contains an external interface that is to be used for multipath load balancing.

Step 6	**neighbor** *ip-address* **dmzlink-bw**
	Example:
	`Router(config-router-af)#`
	`neighbor 172.16.1.1 dmzlink-bw`
		Configures BGP to include the link bandwidth attribute for routes learned from the external interface specified IP address.
		• This command must be configured for each eBGP link that is to be configured as a multipath. Enabling this command allows the bandwidth of the external link to be propagated through the link bandwidth extended community.

Step 7	**neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**] **Example:**	(Optional) Enables community and/or extended community exchange with the specified neighbor.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor 10.10.10.1 send-community extended`	• This command must be configured for iBGP peers to which the link bandwidth extended community attribute is to be propagated.
Step 8	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode, and enters Privileged EXEC mode.

Verifying BGP Link Bandwidth Configuration

To verify the BGP Link Bandwidth feature, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **show ip bgp** *ip-address* [**longer-prefixes** [**injected**] | **shorter-prefixes** [*mask-length*]]
3. **show ip route** [[*ip-address* [*mask*] [**longer-prefixes**]] | [*protocol* [*process-id*]] | [**list** *access-list-number* | *access-list-name*] | [**static download**]]

DETAILED STEPS

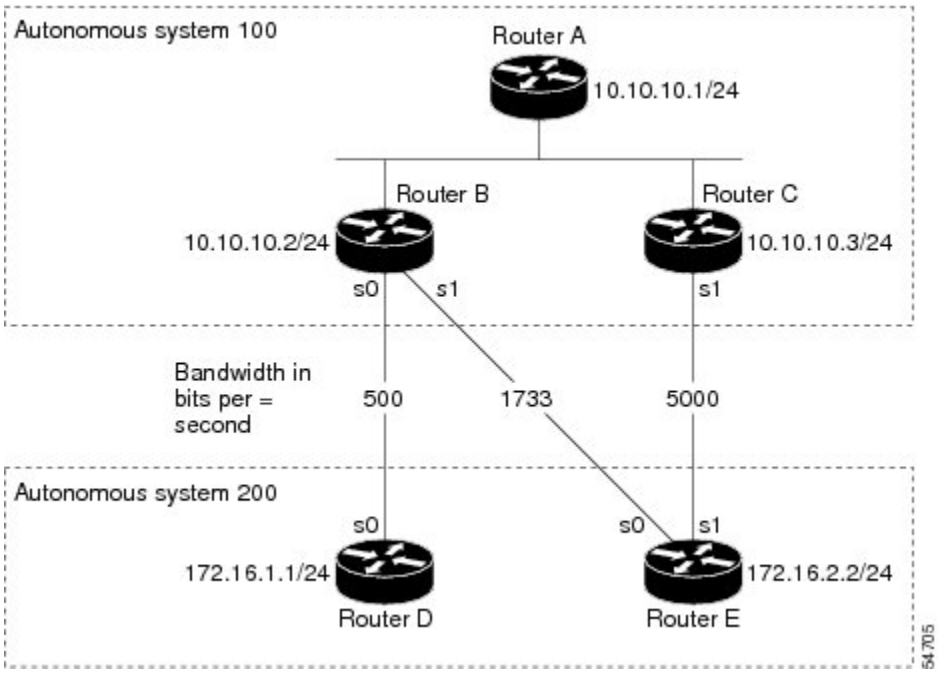
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp** *ip-address* [**longer-prefixes** [**injected**]	**shorter-prefixes** [*mask-length*]] **Example:** `Router# show ip bgp 10.0.0.0`	Displays information about the TCP and BGP connections to neighbors. • The output displays the status of the link bandwidth configuration. The bandwidth of the link is shown in kilobytes.
Step 3	**show ip route** [[*ip-address* [*mask*] [**longer-prefixes**]]	[*protocol* [*process-id*]]	[**list** *access-list-number*	*access-list-name*]	[**static download**]] **Example:** `Router# show ip route 10.0.0.0`	Displays the current state of the routing table. • The output displays traffic share values, including the weights of the links that are used to direct traffic proportionally to the bandwidth of each link.

Configuration Examples for BGP Link Bandwidth

BGP Link Bandwidth Configuration Example

In the following examples, the BGP Link Bandwidth feature is configured so BGP will distribute traffic proportionally to the bandwidth of each external link. The figure below shows two external autonomous systems connected by three links that each carry a different amount of bandwidth (unequal cost links). Multipath load balancing is enabled and traffic is balanced proportionally.

Figure 77: BGP Link Bandwidth Configuration



Router A Configuration

In the following example, Router A is configured to support iBGP multipath load balancing and to exchange the BGP extended community attribute with iBGP neighbors:

```
Router A(config)# router bgp 100

Router A(config-router)# neighbor 10.10.10.2 remote-as 100

Router A(config-router)# neighbor 10.10.10.2 update-source Loopback 0

Router A(config-router)# neighbor 10.10.10.3 remote-as 100

Router A(config-router)# neighbor 10.10.10.3 update-source Loopback 0

Router A(config-router)# address-family ipv4

Router A(config-router)# bgp dmzlink-bw
Router A(config-router-af)# neighbor 10.10.10.2 activate
```

```
Router A(config-router-af)# neighbor 10.10.10.2 send-community both

Router A(config-router-af)# neighbor 10.10.10.3 activate

Router A(config-router-af)# neighbor 10.10.10.3 send-community both

Router A(config-router-af)# maximum-paths ibgp 6
```

Router B Configuration

In the following example, Router B is configured to support multipath load balancing, to distribute Router D and Router E link traffic proportionally to the bandwidth of each link, and to advertise the bandwidth of these links to iBGP neighbors as an extended community:

```
Router B(config)# router bgp 100

Router B(config-router)# neighbor 10.10.10.1 remote-as 100

Router B(config-router)# neighbor 10.10.10.1 update-source Loopback 0

Router B(config-router)# neighbor 10.10.10.3 remote-as 100

Router B(config-router)# neighbor 10.10.10.3 update-source Loopback 0
Router B(config-router)# neighbor 172.16.1.1 remote-as 200

Router B(config-router)# neighbor 172.16.1.1 ebgp-multihop 1
Router B(config-router)# neighbor 172.16.2.2 remote-as 200

Router B(config-router)# neighbor 172.16.2.2 ebgp-multihop 1
Router B(config-router)# address-family ipv4

Router B(config-router-af)# bgp dmzlink-bw

Router B(config-router-af)# neighbor 10.10.10.1 activate

Router B(config-router-af)# neighbor 10.10.10.1 next-hop-self

Router B(config-router-af)# neighbor 10.10.10.1 send-community both

Router B(config-router-af)# neighbor 10.10.10.3 activate

Router B(config-router-af)# neighbor 10.10.10.3 next-hop-self

Router B(config-router-af)# neighbor 10.10.10.3 send-community both

Router B(config-router-af)# neighbor 172.16.1.1
 activate
Router B(config-router-af)# neighbor 172.16.1.1 dmzlink-bw

Router B(config-router-af)# neighbor 172.16.2.2 activate
Router B(config-router-af)# neighbor 172.16.2.2 dmzlink-bw
Router B(config-router-af)# maximum-paths ibgp 6
Router B(config-router-af)# maximum-paths 6
```

Router C Configuration

In the following example, Router C is configured to support multipath load balancing and to advertise the bandwidth of the link with Router E to iBGP neighbors as an extended community:

```
Router C(config)# router bgp 100
Router C(config-router)# neighbor 10.10.10.1 remote-as 100
Router C(config-router)# neighbor 10.10.10.1 update-source Loopback 0
Router C(config-router)# neighbor 10.10.10.2 remote-as 100
Router C(config-router)# neighbor 10.10.10.2 update-source Loopback 0
Router C(config-router)# neighbor 172.16.3.30 remote-as 200
Router C(config-router)# neighbor 172.16.3.30 ebgp-multihop 1
Router C(config-router)# address-family ipv4
Router C(config-router-af)# bgp dmzlink-bw

Router C(config-router-af)# neighbor 10.10.10.1 activate
Router C(config-router-af)# neighbor 10.10.10.1 send-community both
Router C(config-router-af)# neighbor 10.10.10.1 next-hop-self
Router C(config-router-af)# neighbor 10.10.10.2 activate
Router C(config-router-af)# neighbor 10.10.10.2 send-community both
Router C(config-router-af)# neighbor 10.10.10.2 next-hop-self
Router C(config-router-af)# neighbor 172.16.3.3 activate
Router C(config-router-af)# neighbor 172.16.3.3 dmzlink-bw

Router C(config-router-af)# maximum-paths ibgp 6
Router C(config-router-af)# maximum-paths 6
```

Verifying BGP Link Bandwidth

The examples in this section show the verification of this feature on Router A and Router B.

Router B

In the following example, the **show ip bgp** command is entered on Router B to verify that two unequal cost best paths have been installed into the BGP routing table. The bandwidth for each link is displayed with each route.

```
Router B# show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/24, version 48
Paths: (2 available, best #2)
Multipath: eBGP
  Advertised to update-groups:
     1          2
  200
    172.16.1.1 from 172.16.1.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 278 kbytes
  200
    172.16.2.2 from 172.16.2.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 625 kbytes
```

Router A

In the following example, the **show ip bgp** command is entered on Router A to verify that the link bandwidth extended community has been propagated through the iBGP network to Router A. The output shows that a route for each exit link (on Router B and Router C) to autonomous system 200 has been installed as a best path in the BGP routing table.

```
Router A# show ip bgp 192.168.1.0
BGP routing table entry for 192.168.1.0/24, version 48
Paths: (3 available, best #3)
Multipath: eBGP
  Advertised to update-groups:
     1          2
  200
    172.16.1.1 from 172.16.1.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath
      Extended Community: 0x0:0:0
      DMZ-Link Bw 278 kbytes
  200
    172.16.2.2 from 172.16.2.2 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 625 kbytes
  200
    172.16.3.3 from 172.16.3.3 (192.168.1.1)
      Origin incomplete, metric 0, localpref 100, valid, external, multipath, best
      Extended Community: 0x0:0:0
      DMZ-Link Bw 2500 kbytes
```

Router A

In the following example, the **show ip route** command is entered on Router A to verify the multipath routes that are advertised and the associated traffic share values:

```
Router A# show ip route 192.168.1.0
Routing entry for 192.168.1.0/24
  Known via "bgp 100", distance 200, metric 0
  Tag 200, type internal
  Last update from 172.168.1.1 00:01:43 ago
  Routing Descriptor Blocks:
  * 172.168.1.1, from 172.168.1.1, 00:01:43 ago
      Route metric is 0, traffic share count is 13
      AS Hops 1, BGP network version 0
      Route tag 200
    172.168.2.2, from 172.168.2.2, 00:01:43 ago
      Route metric is 0, traffic share count is 30
      AS Hops 1, BGP network version 0
      Route tag 200
    172.168.3.3, from 172.168.3.3, 00:01:43 ago
      Route metric is 0, traffic share count is 120
      AS Hops 1, BGP network version 0
      Route tag 200
```

Additional References

The following sections provide references related to the BGP Link Bandwidth feature.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*

Related Topic	Document Title
BGP multipath load sharing for both eBGP and iBGP in an MPLS-VPN	" BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN"
iBGP multipath load sharing	"iBGP Multipath Load Sharing"
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

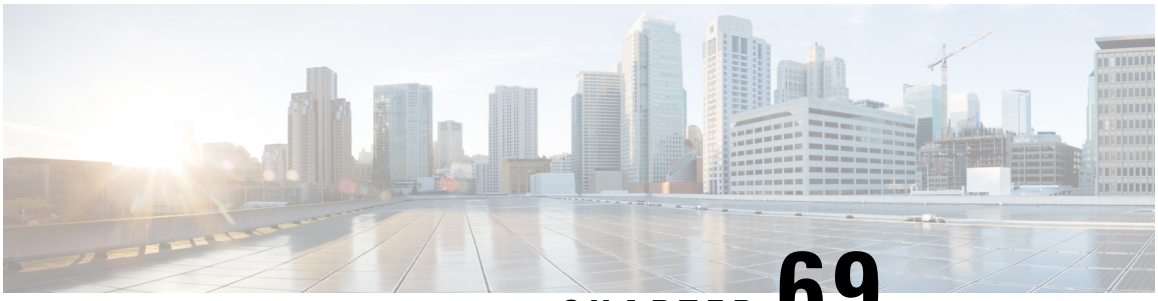
Feature Information for BGP Link Bandwidth

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 94: Feature Information for BGP Link Bandwidth

Feature Name	Releases	Feature Information
BGP Link Bandwidth	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were added or modified by this feature: **bgp dmzlink-bw, neighbor dmzlink-bw**.



CHAPTER 69

Border Gateway Protocol Link-State

Border Gateway Protocol Link-State (BGP-LS) is an Address Family Identifier (AFI) and Sub-address Family Identifier (SAFI) defined to carry interior gateway protocol (IGP) link-state database through BGP routing protocol. BGP-LS delivers network topology information to topology servers and Application Layer Traffic Optimization (ALTO) servers. BGP-LS allows policy-based control to aggregation, information-hiding, and abstraction. BGP-LS supports IS-IS and OSPFv2.

Information About Border Gateway Protocol Link-State

Overview of Link-State Information in Border Gateway Protocol

In a number of environments, a component external to a network is called upon to perform computations based on the network topology and current state of the connections within the network, including Traffic Engineering (TE) information. This information is typically distributed by interior gateway protocol (IGP) routing protocols within the network.

This module describes a mechanism by which link-state (LS) and Traffic Engineering (TE) information from IGPs are collected from networks and shared with external components using the BGP routing protocol, which uses a new BGP Network Layer Reachability Information (NLRI) encoding format. This mechanism is applicable to both physical and virtual links. Applications of this technique include Application-Layer Traffic Optimization (ALTO) servers and Path Computation Elements (PCEs), which are outside the network, but requires real-time information of the state of the network. For example, the link-state database information of each IGP node (OSPF or IS-IS) from the entire network.

In order to address the need for applications that require topological visibility across IGP areas, or even across Autonomous Systems (AS), the BGP-LS address-family or a sub-address-family have been defined to allow BGP to carry link-state information. The identifying key of each link-state object, for example, a node, link, or prefix, is encoded in the NLRI and the properties of the object are encoded in the BGP-LS attribute.

The below figure describes a typical deployment scenario of a network that utilizes BGP-LS. In each IGP area, one or more nodes are configured with BGP-LS. These BGP speakers form an IBGP mesh by connecting to one or more route- reflectors. This way, all BGP speakers (specifically the route- reflectors (RR)) obtain link-state information from all IGP areas (and from other ASes from EBGP peers). An external component connects to the route-reflector to obtain this information (perhaps moderated by a policy regarding what information is or is not advertised to the external component). An external component (for example, a controller) then can collect these information in the "northbound" direction across IGP areas or ASes and construct the end-to-end path (with its associated SIDs) that are applied to an incoming packet for end-to-end forwarding.

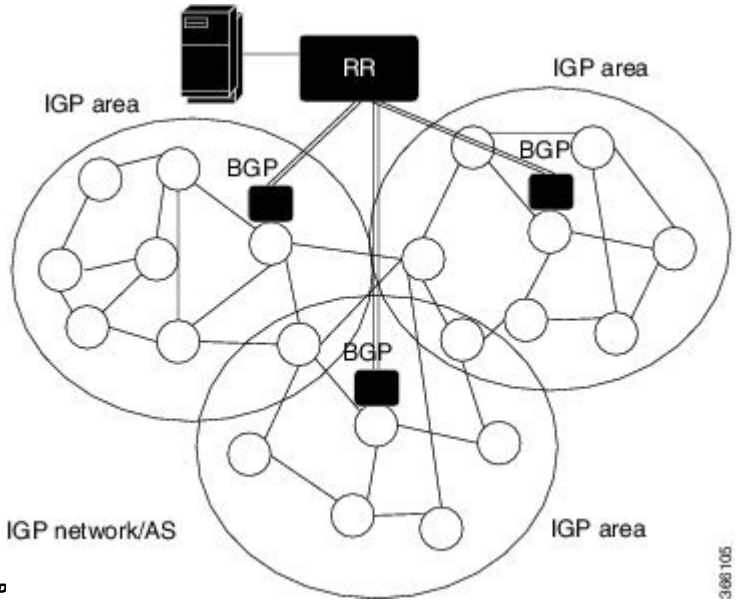


Figure 78: Relation between IGP nodes and BGP

Carrying Link-State Information in Border Gateway Protocol

Carrying link-state information contains two parts:

- Definition of a new BGP NLRI that describes links, nodes, and prefixes comprises of IGP link-state information.
- Definition of a new BGP-LS attribute that carries link, node, and prefix properties and attributes, such as the link and prefix metric or auxiliary Router IDs of nodes, and so on.

TLV Format

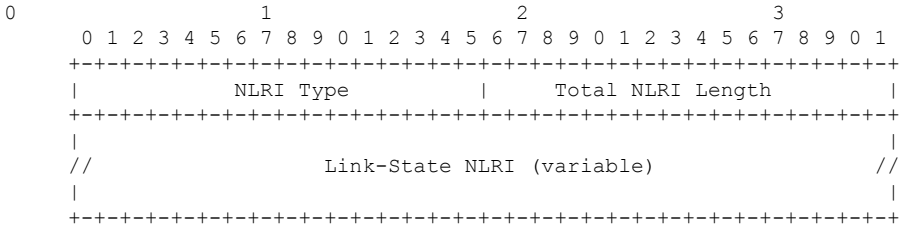
Information in the new Link-State NLRIs and attributes is encoded in Type/Length/Value (TLV) triplets. The TLV format is shown in the below figure.

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                          Value (variable)                     //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Length field defines the length of the value portion in octets (thus, a TLV with no value portion would have a length of zero).

Link-State NLRI

The MP_REACH_NLRI and MP_UNREACH_NLRI attributes are BGP's containers for carrying opaque information. Each Link-State Network Layer Reachability Information (NLRI) describes either a node, a link, or a prefix. NLRI body is a set of Type/Length/Value triplets (TLV) and contains the data that identifies an object.



NLRI Types

The Total NLRI length field contains the cumulative length, in octets, of the rest of the NLRI, not including the NLRI Type field or itself.

Figure 79: The NLRI Types

Type	NLRI Type
1	Node NLRI
2	Link NLRI
3	IPv4 Topology Prefix NLRI
4	IPv6 Topology Prefix NLRI

The NLRI Types are shown in the following figures:

Figure 80: The Node NLRI Format

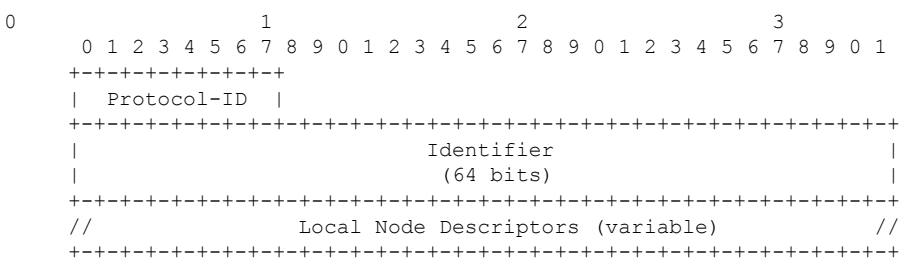
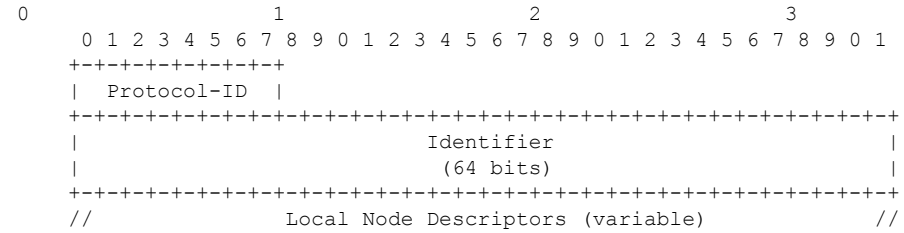
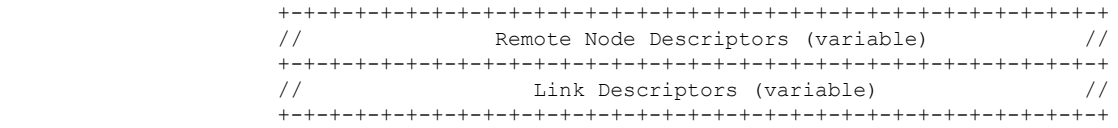


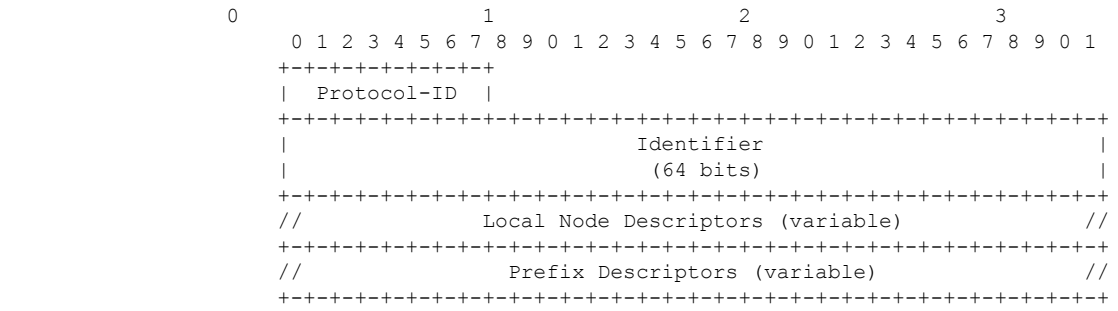
Figure 81: The Link NLRI Format





The IPv4 and IPv6 Prefix NLRIs (NLRI Type = 3 and Type = 4) use the same format, as shown in the following figure.

Figure 82: The IPv4/IPv6 Topology Prefix NLRI Format



Node Descriptors

Each link is anchored by a pair of Router-IDs that are used by the underlying IGP, namely, a 48-bit ISO System-ID for IS-IS and a 32-bit Router-ID for OSPFv2 and OSPFv3. An IGP may use one or more additional auxiliary Router-IDs, mainly for traffic engineering purposes. For example, IS-IS may have one or more IPv4 and IPv6 TE Router-IDs. These auxiliary Router-IDs must be included in the link attribute.

Link Descriptors

The Link Descriptor field is a set of Type/Length/Value (TLV) triplets. The link descriptor TLVs uniquely identify a link among multiple parallel links between a pair of anchor routers. A link described by the link descriptor TLVs actually is a "half-link", a unidirectional representation of a logical link. In order to fully describe a single logical link, two originating routers advertise a half-link each, that is, two Link NLRIs are advertised for a given point-to-point link.

Prefix Descriptors

The Prefix Descriptor field is a set of Type/Length/Value (TLV) triplets. Prefix Descriptor TLVs uniquely identify an IPv4 or IPv6 prefix originated by a node.

BGP-LS Attribute

The BGP-LS attribute is an optional, non-transitive BGP attribute that is used to carry link, node, and prefix parameters and attributes. It is defined as a set of Type/Length/Value (TLV) triplets. This attribute should only be included with Link-State NLRIs. This attribute must be ignored for all other address families.

How to Configure OSPF With Border Gateway Protocol Link-State

OSPF is one of the IGP protocols that feeds its topology into BGP into the LS cache. Link state information can be passed to BGP in two ways:

- When new communications between OSPF and BGP has been established, or when BGP-LS functionality has been initially enabled under OSPF, then all LSA information is downloaded to BGP via the LS library.
- As new LSA information is being processed or received from remote OSPF nodes, this information is added or updated in BGP.

Configuring Border Gateway Protocol Link-State With OSPF

Perform the following steps to configure OSPF with BGP-LS:

1. Enable the OSPF routing protocol and enter router configuration mode.

   ```
   router ospf
   ```

   For example,

   ```
   Device(config-router)# router ospf 10
   ```

2. Distribute BGP link-state.

   ```
    distribute link-state
   ```

   For example,

   ```
   Device(config-router)# distribute link-state instance-id <instid>
   ```

   ```
   Device(config-router)# distribute link-state throttle <time>
   ```

   instance-id (optional): Sets instance ID for LS distribution. Default Value is 0. Range: 32 to 2^32-1.

   throttle (optional): Sets throttle time to process LS distribution queue. Default value is 5 seconds. Range: 1 to 3600 seconds.

Note In the scenarios where any area gets deleted, throttle timer does not get honored. Queue is walked by OSPF completely and updates to all the areas are sent to BGP.

If you do not specify any value for instance ID and throttle, default values are taken.

Example:

```
#show run | sec router ospf
router ospf 10
distribute link-state instance-id 33 throttle 6
```

Note You should not be using the same instance ID for two OSPF instances. It throws an instance ID already in use error.

How to Configure IS-IS With Border Gateway Protocol Link-State

IS-IS distributes routing information into BGP. IS-IS processes the routing information in it's LSP database and extract the relevant objects. It advertises IS-IS nodes, links, and prefix information and their attributes into BGP. This update from IS-IS into BGP only happens when there is a change in the LSP fragments, either belonging to the local router or any remote routers.

Configuring IS-IS With Border Gateway Protocol Link-State

Perform the following steps to configure IS-IS with BGP-LS:

1. Enable the IS-IS routing protocol and enter router configuration mode.

   ```
   router isis
   ```

   For example,

   ```
   Device(config-router)# router isis
   ```

2. Distribute BGP link-state.

   ```
    distribute link-state
   ```

   For example,

   ```
   Device(config-router)# distribute link-state instance-id <instid>
   Device(config-router)# distribute link-state throttle <time>
   ```

   instance-id (optional): Sets instance ID for LS distribution. The range is from 32-4294967294.

   throttle (optional): Sets throttle time to process LS distribution queue. The range is from 5-20 seconds.

Configuring BGP

Perform the following steps to configure BGP with BGP-LS:

1. Enable the BGP routing protocol and enter router configuration mode.

   ```
   router bgp
   ```

   For example,

   ```
   Device(config-if)# router bgp 100
   ```

2. Configure the address-family link-state.

   ```
   address-family link-state link-state
   ```

For example,

```
Device(config-router)# address-family link-state link-state
```

3. Exit the address-family.

```
exit-address-family
```

For example,

```
Device(config-router)# exit-address-family
```

Example: Configuring ISIS With Border Gateway Protocol Link-State

Example: IS-IS Configuration

```
router isis 1
 net 49.0001.1720.1600.1001.00
 is-type level-1
 metric-style wide
 distribute link-state level-1
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1

interface GigabitEthernet2/2/2
 ip address 172.16.0.1 255.255.0.0
 ip router isis 1
 negotiation auto
 mpls traffic-eng tunnels
 isis network point-to-point
```

Example: BGP Configuration

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 10.0.0.1 remote-as 100
 neighbor 10.0.0.4 remote-as 100
 !
 address-family ipv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.4 activate
 exit-address-family
 !
 address-family link-state link-state
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.4 activate
 exit-address-family
```

Verifying Border Gateway Protocol Link-State Configurations

Use the following show commands in any order to verify the status of the BGP-LS configurations.

show ip ospf ls-distribution

Displays the status of LS distribution.

```
Device# show ip ospf ls-distribution

            OSPF Router with ID (10.0.0.6) (Process ID 10)
          OSPF LS Distribution is Enabled
                        Instance Id: 0
                     Throttle time: 5
                  Registration Handle: 0x0
                               Status:Ready Active
    Num DBs Queued for LSCache Update: 0
     Num of DBs with Unresolved Links: 0
```

show ip ospf database dist-ls-pending

Displays the LSAs that are pending, to be sent to BGP.

```
 Sample Output:
Device# show ip ospf database dist-ls-pending

            OSPF Router with ID (10.0.0.6) (Process ID 10)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
 10.0.0.7         10.0.0.6         4           0x80000006 0x009678 1
 172.16.0.6         172.16.0.6         1110        0x80000018 0x00CAF9 2
(Has-unresolved-links)
```

show isis distribute-ls [level-1 | level-2]

Displays IS-IS internal LS cache information that are distributed to BGP.

```
Device# sh isis distribute-ls

 ISIS distribute link-state: configured
 distls_levels:0x3, distls_initialized:1,
 distls_instance_id:0, distls_throttle_delay:10
 LS DB: ls_init_started(0) ls_initialized(1) ls_pending_delete(0)
 distls_enabled[1]:1
 distls_enabled[2]:1
 Level 1:
Node System ID:0003.0003.0003 Pseudonode-Id:0 ls_change_flags:0x0
   LSP: lspid(0003.0003.0003.00-00), lsptype(0) lsp_change_flags(0x0)
    Node Attr: name(r3) bitfield(0xD1) node_flags(0x0)
     area_len/area_addr(2/33) num_mtid/mtid(0/0) ipv4_id(172.16.0.9)
     num_alg/sr_alg(0/0) num_srgb/srgb(1/(start:16000, range:8000)
     srgb_flags(0x80)
     opaqe_len/opaqe(0/0x0)
    ISIS LS Links:
     mtid(0): nid:0002.0002.0002.00, {0, 0}, {6.6.6.1, 6.6.6.6}
     Link Attr: bitbfield:0x940F, local_ipv4_id:6.6.6.1, remote_ipv4_id:172.16.0.8,
      max_link_bw:10000, max_resv_bw:10000,
     num_unresv_bw/unresv_bw:8/
      [0]:    10000 kbits/sec, [1]:     8000 kbits/sec
      [2]:     8000 kbits/sec, [3]:     8000 kbits/sec
      [4]:     8000 kbits/sec, [5]:     8000 kbits/sec
      [6]:     8000 kbits/sec, [7]:     8000 kbits/sec,
      admin_group:0, protect_type:0, mpls_proto_mask:0x0,
      te_metric:0, metric:0, link_name:,
      num_srlg/srlg:0/
      num_adj_sid/adjsid:2/
    Adjacency SID Label:16 F:0 B:0 V:1 L:1 S:0 weight:0
    Adjacency SID Label:17 F:0 B:1 V:1 L:1 S:0 weight:0
      opaque_len/opaque_data:0/0x0
    Address-family ipv4 ISIS LS Prefix:
```

```
     mtid(0): 1.1.1.0/24
     Prefix Attr: bitfield:0x0, metric:10, igp_flags:0x0,
      num_route_tag:0, route_tag:0
      num_pfx_sid:0, pfx_sid:
      pfx_srms:
      opaque_len:0, opaque_data:0x0
     mtid(0): 172.16.0.8/24
     Prefix Attr: bitfield:0x0, metric:10, igp_flags:0x0,
      num_route_tag:0, route_tag:0
      num_pfx_sid:0, pfx_sid:
      pfx_srms:
      opaque_len:0, opaque_data:0x0
```

show bgp link-state link-state

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Prefix codes:E link, V node, T4 IPv4 reachable route, T6 IPv6 reachable route, I Identifier,

              N local node, R remote node, L link, P prefix,
              L1/L2 ISIS level-1/level-2, O OSPF, a area-ID, l link-ID,
              t topology-ID, s ISO-ID, c confed-ID/ASN, b bgp-identifier,
              r router-ID, i if-address, n nbr-address, o OSPF Route-type,
              p IP-prefix, d designated router address, u/U Unknown,
              x/X Unexpected, m/M Malformed

     Network          Next Hop            Metric LocPrf Weight Path
 *>   [V][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.1001.00]]
                      15.0.0.1                 0             0 100 i
 *>   [V][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]]
                      15.0.0.1                 0             0 100 i
 *>   [V][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.3003.00]]
                      15.0.0.1                 0             0 100 i
 *>   [V][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.4004.00]]
                      15.0.0.1                 0             0 100 i
 *>   [V][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.5005.00]]
                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.1001.00]][R[c100][b0.0.0.0][s1720.1600.2002.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]][R[c100][b0.0.0.0][s1720.1600.1001.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]][R[c100][b0.0.0.0][s1720.1600.3003.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]][R[c100][b0.0.0.0][s1720.1600.4004.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.3003.00]][R[c100][b0.0.0.0][s1720.1600.2002.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.3003.00]][R[c100][b0.0.0.0][s1720.1600.5005.00]][L]
```

```
                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.4004.00]][R[c100][b0.0.0.0][s1720.1600.2002.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.4004.00]][R[c100][b0.0.0.0][s1720.1600.5005.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.5005.00]][R[c100][b0.0.0.0][s1720.1600.3003.00]][L]

                      15.0.0.1                 0             0 100 i
 *>
[E][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.5005.00]][R[c100][b0.0.0.0][s1720.1600.4004.00]][L]

                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.1001.00]][P[p10.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.1001.00]][P[p7.7.7.7/32]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]][P[p10.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]][P[p11.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]][P[p12.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.2002.00]][P[p5.5.5.5/32]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.3003.00]][P[p11.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.3003.00]][P[p13.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.3003.00]][P[p3.3.3.3/32]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.4004.00]][P[p12.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.4004.00]][P[p14.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.4004.00]][P[p15.15.15.15/32]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.5005.00]][P[p13.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.5005.00]][P[p14.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.5005.00]][P[p15.0.0.0/24]]
                      15.0.0.1                 0             0 100 i
 *>   [T4][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.5005.00]][P[p16.16.16.16/32]]
                      15.0.0.1                 0             0 100 i
```

show bgp link-state link-state nlri <nlri string>

```
BGP routing table entry for [V][L1][I0x43][N[c100][b0.0.0.0][s1720.1600.4004.00]], version
 95
Paths: (1 available, best #1, table link-state link-state)
  Not advertised to any peer
  Refresh Epoch 4
  Local
    16.16.16.16 (metric 30) from 15.15.15.15 (15.15.15.15)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 16.16.16.16, Cluster list: 15.15.15.15
      LS Attribute: Node-name: R4, ISIS area: 49.12.34
      rx pathid: 0, tx pathid: 0x0
```

Border Gateway Protocol Link-State Debug Commands

- **debug ip ospf dist-ls [detail]**

  Turns on ls-distribution related debugs in OSPF.

- **debug isis distribute-ls**

  Displays the items being advertised into the BGP from IS-IS.

Additional References for Border Gateway Protocol Link-State

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

MIBs

MIB	MIBs Link
• CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
• RFC 7752	*Link-State Info Distribution Using BGP*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

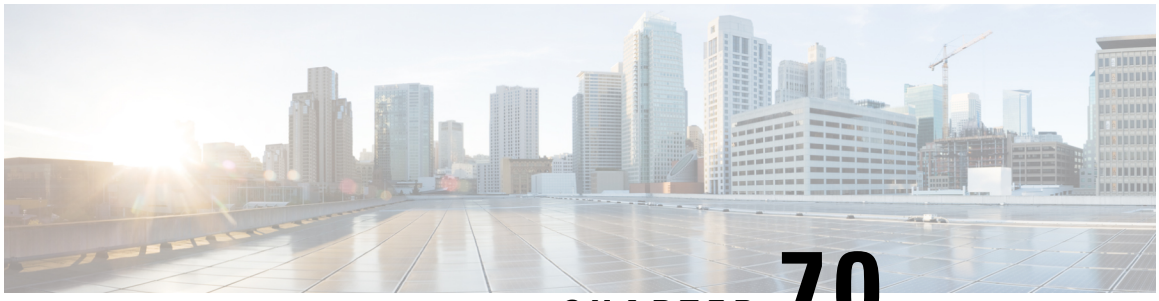
Feature Information for Border Gateway Protocol Link-State

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 95: Feature Information for BGP-LS

Feature Name	Releases	Feature Information
Border Gateway Protocol Link-State	Cisco IOS XE Everest 16.4.1	BGP Link-State (LS) is an Address Family Identifier (AFI) and Sub-address Family Identifier (SAFI) defined to carry interior gateway protocol (IGP) link-state database through BGP. The following commands were introduced or modified: **address-family link-state link-state**, **distribute link-state**, **show bgp link-state link-state**, **show bgp link-state link-state nlri** *nlri string*, **show ip ospf database dist-ls-pending**, **show ip ospf ls-distribution**, **show isis distribute-ls**



CHAPTER 70

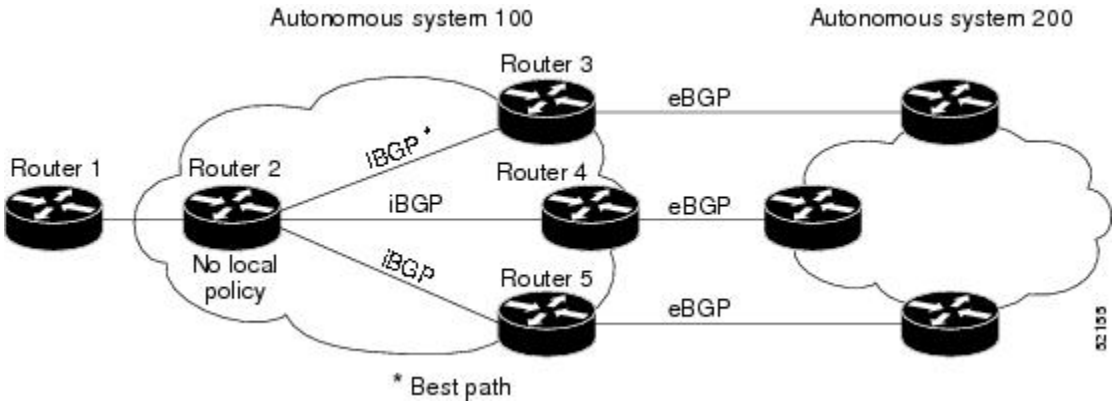
iBGP Multipath Load Sharing

This feature module describes the iBGP Multipath Load Sharing feature. This feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination. The best paths or multipaths are then installed in the IP routing table of the router.

iBGP Multipath Load Sharing Overview

When a Border Gateway Protocol (BGP) speaking router with no local policy configured receives multiple network layer reachability information (NLRI) from the internal BGP (iBGP) for the same destination, the router will choose one iBGP path as the best path. The best path is then installed in the IP routing table of the router. For example, in the figure below, although there are three paths to autonomous system 200, Router 2 determines that one of the paths to autonomous system 200 is the best path and uses this path only to reach autonomous system 200.

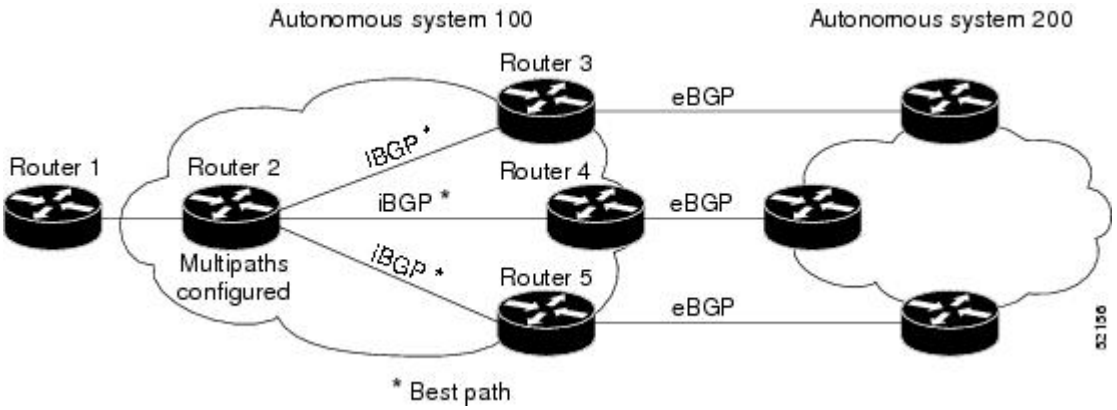
Figure 83: Non-MPLS Topology with One Best Path



The iBGP Multipath Load Sharing feature enables the BGP speaking router to select multiple iBGP paths as the best paths to a destination. The best paths or multipaths are then installed in the IP routing table of the router. For example, on router 2 in the figure below, the paths to routers 3, 4, and 5 are configured as multipaths

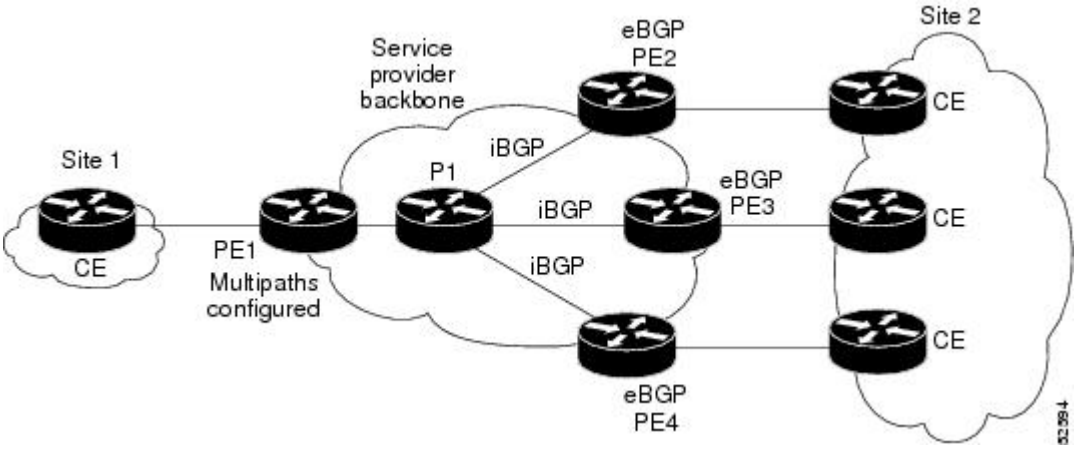
and can be used to reach autonomous system 200, thereby equally sharing the load to autonomous system 200.

Figure 84: Non-MPLS Topology with Three Multipaths



The iBGP Multipath Load Sharing feature functions similarly in a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) with a service provider backbone. For example, on router PE1 in the figure below, the paths to routers PE2, PE3, and PE4 can be selected as multipaths and can be used to equally share the load to site 2.

Figure 85: MPLS VPN with Three Multipaths



For multiple paths to the same destination to be considered as multipaths, the following criteria must be met:

- All attributes must be the same. The attributes include weight, local preference, autonomous system path (entire attribute and not just length), origin code, Multi Exit Discriminator (MED), and Interior Gateway Protocol (IGP) distance.
- The next hop router for each multipath must be different.

Even if the criteria are met and multiple paths are considered multipaths, the BGP speaking router will still designate one of the multipaths as the best path and advertise this best path to its neighbors.

Benefits of iBGP Multipath Load Sharing

Configuring multiple iBGP best paths enables a router to evenly share the traffic destined for a particular site.

Restrictions on iBGP Multipath Load Sharing

Route Reflector Limitation

With multiple iBGP paths installed in a routing table, a route reflector will advertise only one of the paths (one next hop).

Memory Consumption Restriction

Each IP routing table entry for a BGP prefix that has multiple iBGP paths uses approximately 350 bytes of additional memory. We recommend not using this feature on a router with a low amount of available memory and especially when the router is carrying a full Internet routing table.

How to Configure iBGP Multipath Load Sharing

Configuring iBGP Multipath Load Sharing

To configure the iBGP Multipath Load Sharing feature, use the following command in router configuration mode:

Command	Purpose
`Device(config-router)# maximum-paths ibgp maximum-number`	Controls the maximum number of parallel iBGP routes that can be installed in a routing table.

Verifying iBGP Multipath Load Sharing

To verify that the iBGP Multipath Load Sharing feature is configured correctly, perform the following steps:

SUMMARY STEPS

1. Enter the **show ip bgp** *network-number* EXEC command to display attributes for a network in a non-MPLS topology, or the **show ip bgp vpnv4 all** *ip-prefix* EXEC command to display attributes for a network in an MPLS VPN:
2. In the display resulting from the **show ip bgp** *network-number* EXEC command or the **show ip bgp vpnv4 all** *ip-prefix* EXEC command, verify that the intended multipaths are marked as “multipaths." Notice that one of the multipaths is marked as “best.”
3. Enter the **show ip route** *ip-address* EXEC command to display routing information for a network in a non-MPLS topology or the **show ip route vrf** *vrf-name ip-prefix* EXEC command to display routing information for a network in an MPLS VPN:

4. Verify that the paths marked as "multipath" in the display resulting from the **show ip bgp** *ip-prefix* EXEC command or the **show ip bgp vpnv4 all** *ip-prefix* EXEC command are included in the routing information. (The routing information is displayed after performing Step 3.)

DETAILED STEPS

Step 1 Enter the **show ip bgp** *network-number* EXEC command to display attributes for a network in a non-MPLS topology, or the **show ip bgp vpnv4 all** *ip-prefix* EXEC command to display attributes for a network in an MPLS VPN:

Example:

```
Device# show ip bgp 10.22.22.0

BGP routing table entry for 10.22.22.0/24, version 119
Paths:(6 available, best #1)
Multipath:iBGP
Flag:0x820
  Advertised to non peer-group peers:
  10.1.12.12
  22
    10.2.3.8 (metric 11) from 10.1.3.4 (100.0.0.5)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
      Originator:100.0.0.5, Cluster list:100.0.0.4
  22
    10.2.1.9 (metric 11) from 10.1.1.2 (100.0.0.9)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Originator:100.0.0.9, Cluster list:100.0.0.2
  22
    10.2.5.10 (metric 11) from 10.1.5.6 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Originator:100.0.0.10, Cluster list:100.0.0.6
  22
    10.2.4.10 (metric 11) from 10.1.4.5 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Originator:100.0.0.10, Cluster list:100.0.0.5
  22
    10.2.6.10 (metric 11) from 10.1.6.7 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Originator:100.0.0.10, Cluster list:100.0.0.7

Device# show ip bgp vpnv4 all 10.22.22.0

BGP routing table entry for 100:1:10.22.22.0/24, version 50
Paths:(6 available, best #1)
Multipath:iBGP
  Advertised to non peer-group peers:
  200.1.12.12
  22
    10.22.7.8 (metric 11) from 10.11.3.4 (100.0.0.8)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath, best
      Extended Community:RT:100:1
      Originator:100.0.0.8, Cluster list:100.1.1.44
  22
    10.22.1.9 (metric 11) from 10.11.1.2 (100.0.0.9)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1
      Originator:100.0.0.9, Cluster list:100.1.1.22
  22
    10.22.6.10 (metric 11) from 10.11.6.7 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
```

```
      Extended Community:RT:100:1
      Originator:100.0.0.10, Cluster list:100.0.0.7
  22
    10.22.4.10 (metric 11) from 10.11.4.5 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1
      Originator:100.0.0.10, Cluster list:100.0.0.5
  22
    10.22.5.10 (metric 11) from 10.11.5.6 (100.0.0.10)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1
      Originator:100.0.0.10, Cluster list:100.0.0.6
```

Step 2 In the display resulting from the **show ip bgp** *network-number* EXEC command or the **show ip bgp vpnv4 all** *ip-prefix* EXEC command, verify that the intended multipaths are marked as “multipaths." Notice that one of the multipaths is marked as “best.”

Step 3 Enter the **show ip route** *ip-address* EXEC command to display routing information for a network in a non-MPLS topology or the **show ip route vrf** *vrf-name ip-prefix* EXEC command to display routing information for a network in an MPLS VPN:

Example:

```
Device# show ip route 10.22.22.0

Routing entry for 10.22.22.0/24
  Known via "bgp 1", distance 200, metric 0
  Tag 22, type internal
  Last update from 10.2.6.10 00:00:03 ago
  Routing Descriptor Blocks:
  * 10.2.3.8, from 10.1.3.4, 00:00:03 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.2.1.9, from 10.1.1.2, 00:00:03 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.2.5.10, from 10.1.5.6, 00:00:03 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.2.4.10, from 10.1.4.5, 00:00:03 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.2.6.10, from 10.1.6.7, 00:00:03 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1

Device# show ip route vrf PATH 10.22.22.0

Routing entry for 10.22.22.0/24
  Known via "bgp 1", distance 200, metric 0
  Tag 22, type internal
  Last update from 10.22.5.10 00:01:07 ago
  Routing Descriptor Blocks:
  * 10.22.7.8 (Default-IP-Routing-Table), from 10.11.3.4, 00:01:07 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.22.1.9 (Default-IP-Routing-Table), from 10.11.1.2, 00:01:07 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.22.6.10 (Default-IP-Routing-Table), from 10.11.6.7, 00:01:07 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.22.4.10 (Default-IP-Routing-Table), from 10.11.4.5, 00:01:07 ago
```

```
  Route metric is 0, traffic share count is 1
  AS Hops 1
10.22.5.10 (Default-IP-Routing-Table), from 10.11.5.6, 00:01:07 ago
  Route metric is 0, traffic share count is 1
  AS Hops 1
```

Step 4 Verify that the paths marked as "multipath" in the display resulting from the **show ip bgp** *ip-prefix* EXEC command or the **show ip bgp vpnv4 all** *ip-prefix* EXEC command are included in the routing information. (The routing information is displayed after performing Step 3.)

Monitoring and Maintaining iBGP Multipath Load Sharing

To display iBGP Multipath Load Sharing information, use the following commands in EXEC mode, as needed:

Command	Purpose
`Device# show ip bgp ip-prefix`	Displays attributes and multipaths for a network in a non-MPLS topology.
`Device# show ip bgp vpnv4 all ip-prefix`	Displays attributes and multipaths for a network in an MPLS VPN.
`Device# show ip route ip-prefix`	Displays routing information for a network in a non-MPLS topology.
`Device# show ip route vrf vrf-name ip-prefix`	Displays routing information for a network in an MPLS VPN.

Configuration Examples

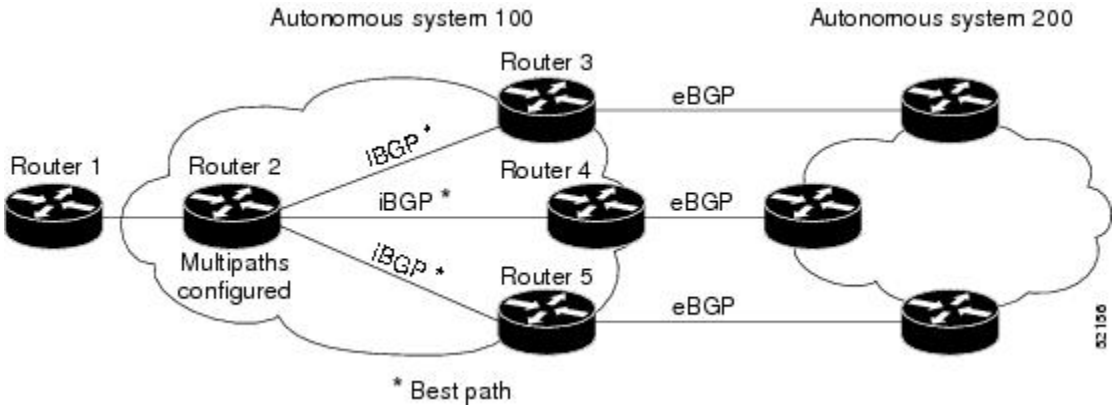
Both examples assume that the appropriate attributes for each path are equal and that the next hop router for each multipath is different.

Example: iBGP Multipath Load Sharing in a Non-MPLS Topology

Both examples assume that the appropriate attributes for each path are equal and that the next hop router for each multipath is different.

The following example shows how to set up the iBGP Multipath Load Sharing feature in a non-MPLS topology (see the figure below).

Figure 86: Non-MPLS Topology Example



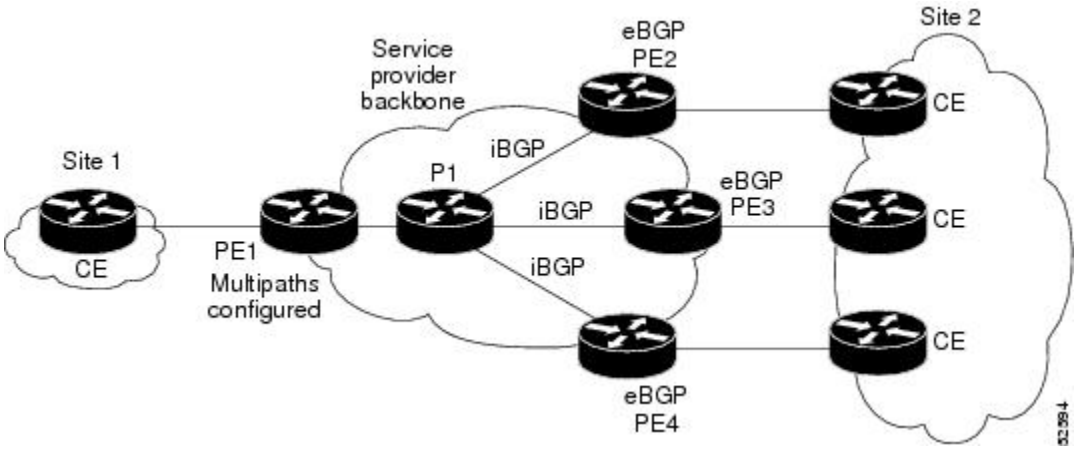
Router 2 Configuration

```
router bgp 100
maximum-paths ibgp 3
```

Example: iBGP Multipath Load Sharing in an MPLS VPN Topology

The following example shows how to set up the iBGP Multipath Load Sharing feature in an MPLS VPN topology (see the figure below).

Figure 87: MPLS VPN Topology Example



Router PE1 Configuration

```
router bgp 100
address-family ipv4 unicast vrf site2
 maximum-paths ibgp 3
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP multipath load sharing for both eBGP and iBGP in an MPLS-VPN	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN" module in the *IP Routing: BGP Configuration Guide*
Advertising the bandwidth of an autonomous system exit link as an extended community	"BGP Link Bandwidth" module in the *IP Routing: BGP Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

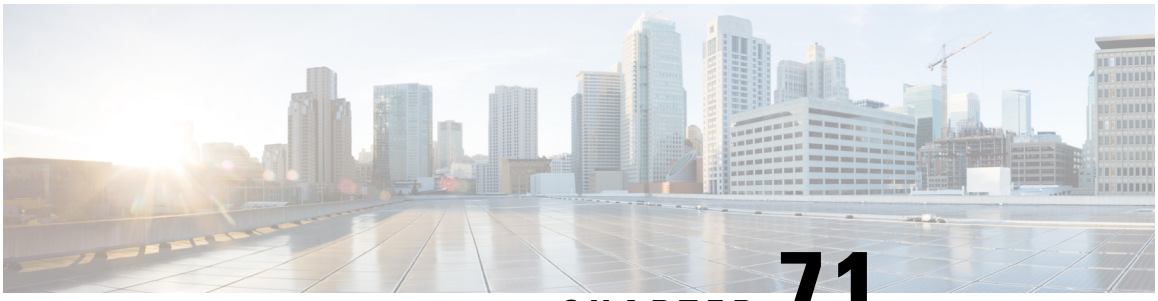
Feature Information for iBGP Multipath Load Sharing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 96: Feature Information for iBGP Multipath Load Sharing

Feature Name	Releases	Feature Information
iBGP multipath load sharing	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were modified by this feature: **maximum paths ibgp**, **show ip bgp**, **show ip bgp vpnv4**, **show ip route**, **show ip route vrf**.



CHAPTER 71

BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The BGP Multipath Load Sharing for both eBGP and iBGP in an MPLS-VPN feature allows you to configure multipath load balancing with both external BGP (eBGP) and internal BGP (iBGP) paths in Border Gateway Protocol (BGP) networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). This feature provides improved load balancing deployment and service offering capabilities and is useful for multihomed autonomous systems and Provider Edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks.

Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Load Balancing is Configured Under CEF

Cisco Express Forwarding (CEF) or distributed CEF (dCEF) must be enabled on all participating routers.

Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Address Family Support

This feature is configured on a per VPN routing and forwarding instance (VRF) basis. This feature can be configured under only the IPv4 VRF address family.

Memory Consumption Restriction

Each BGP multipath routing table entry will use additional memory. We recommend that you do not use this feature on a router with a low amount of available memory and especially if router is carries full Internet routing tables.

Route Reflector Limitation

When multiple iBGP paths installed in a routing table, a route reflector will advertise only one paths (next hop). If a router is behind a route reflector, all routers that are connected to multihomed sites will not be advertised unless a different route distinguisher is configured for each VRF.

Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Multipath Load Sharing Between eBGP and iBGP

A BGP routing process will install a single path as the best path in the routing information base (RIB) by default. The **maximum-paths** command allows you to configure BGP to install multiple paths in the RIB for multipath load sharing. BGP uses the best path algorithm to still select a single multipath as the best path and advertise the best path to BGP peers.

Note The number of paths of multipaths that can be configured is documented on the **maximum-paths** command reference page.

Load balancing over the multipaths is performed by CEF. CEF load balancing is configured on a per-packet round robin or on a per session (source and destination pair) basis. For information about CEF, refer to the "Cisco Express Forwarding Overview" documentation:

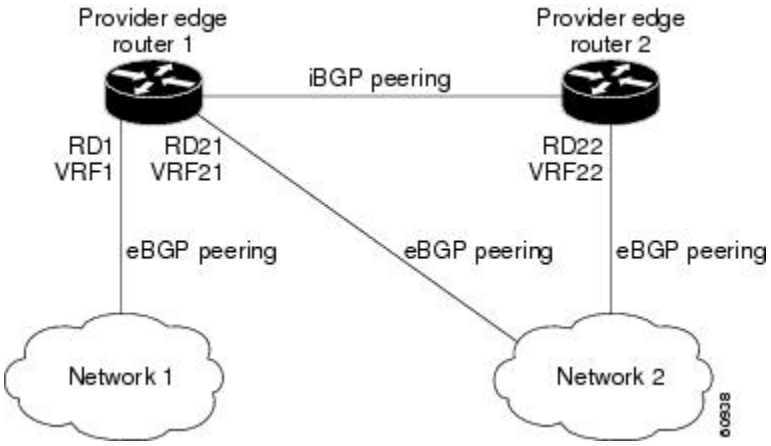
The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature is enabled only under the IPv4 VRF address family configuration mode. When enabled, this feature can perform load balancing on eBGP and/or iBGP paths that are imported into the VRF. The number of multipaths is configured on a per VRF basis. Separate VRF multipath configurations are isolated by unique route distinguisher.

Note The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature operates within the parameters of configured outbound routing policy.

eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network

The figure below shows a service provider BGP MPLS network that connects two remote networks to PE router 1 and PE router 2. PE router 1 and PE router 2 are both configured for VPNv4 unicast iBGP peering. Network 2 is a multihomed network that is connected to PE router 1 and PE router 2. Network 2 also has extranet VPN services configured with Network 1. Both Network 1 and Network 2 are configured for eBGP peering with the PE routers.

Figure 88: A Service Provider BGP MPLS Network

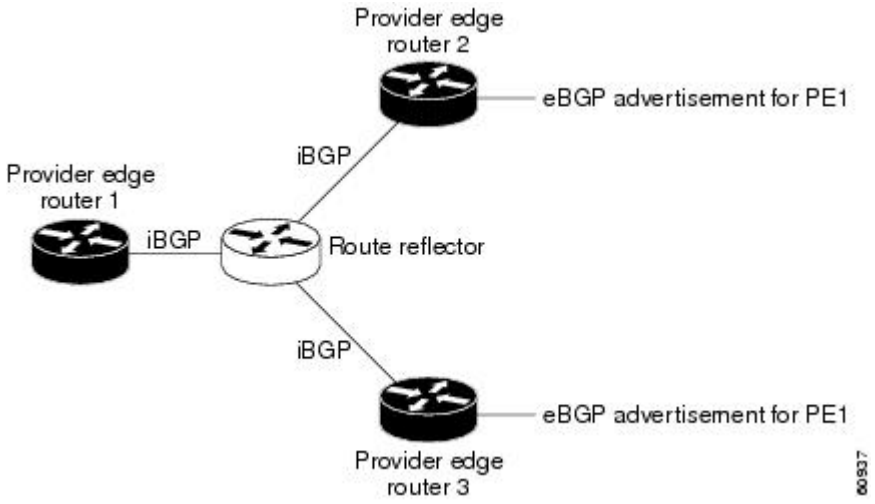


PE router 1 can be configured with the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature so that both iBGP and eBGP paths can be selected as multipaths and imported into the VRF of Network 1. The multipaths will be used by CEF to perform load balancing. IP traffic that is sent from Network 2 to PE router 1 and PE router 2 will be sent across the eBGP paths as IP traffic. IP traffic that is sent across the iBGP path will be sent as MPLS traffic, and MPLS traffic that is sent across an eBGP path will be sent as IP traffic. Any prefix that is advertised from Network 2 will be received by PE router 1 through route distinguisher (RD) 21 and RD 22.The advertisement through RD 21 will be carried in IP packets, and the advertisement through RD 22 will be carried in MPLS packets. Both paths can be selected as multipaths for VRF1 and installed into the VRF1 RIB.

eBGP and iBGP Multipath Load Sharing With Route Reflectors

The figure below shows a topology that contains three PE routers and a route reflector, all configured for iBGP peering. PE router 2 and PE router 3 each advertise an equal preference eBGP path to PE router 1. By default, the route reflector will choose only one path and advertise PE router 1.

Figure 89: A Topology with a Route Reflector



For all equal preference paths to PE router 1 to be advertised through the route reflector, you must configure each VRF with a different RD. The prefixes received by the route reflector will be recognized differently and advertised to PE router 1.

Benefits of Multipath Load Sharing for Both eBGP and iBGP

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature allows multihomed autonomous systems and PE routers to be configured to distribute traffic across both eBGP and iBGP paths.

HOw to COnfigure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Configuring Multipath Load Sharing for Both eBGP and iBGP

To configure this feature, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **maximum-paths eibgp** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf RED`	Places the router in address family configuration mode. • Separate VRF multipath configurations are isolated by unique route distinguisher.
Step 5	**maximum-paths eibgp** *number* **Example:** `Device(config-router-af)# maximum-paths eibgp 6`	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. **Note** The **maximum-paths eibgp** command can be configured only under the IPv4 VRF address family configuration mode and cannot be configured in any other address family configuration mode.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode, and enters privileged EXEC mode.

Verifying Multipath Load Sharing for Both eBGP an iBGP

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** [*neighbor-address* [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** [*regexp*] | **received** *prefix-filter* | **received-routes** | **routes**]]
3. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*}
4. **show ip route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp neighbors** [*neighbor-address* [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** [*regexp*] | **received** *prefix-filter* | **received-routes** | **routes**]] **Example:** `Device# show ip bgp neighbors`	Displays information about the TCP and BGP connections to neighbors.
Step 3	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} **Example:** `Device# show ip bgp vpnv4 vrf RED`	Displays VPN address information from the BGP table. This command is used to verify that the VRF has been received by BGP.
Step 4	**show ip route vrf** *vrf-name* **Example:** `Device# show ip route vrf RED`	Displays the IP routing table associated with a VRF instance. The show ip route vrf command is used to verify that the VRF is in the routing table.

Configuration Examples for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Example: Configuring eBGP and iBGP Multipath Load Sharing

This following configuration example configures a router in address-family mode to select six BGP routes (eBGP or iBGP) as multipaths:

```
Device(config)# router bgp 40000
 Device(config-router)# address-family ipv4 vrf RED
 Device(config-router-af)# maximum-paths eibgp 6
 Device(config-router-af)# end
```

Example: Verifying eBGP and iBGP Multipath Load Sharing

To verify that iBGP and eBGP routes have been configured for load sharing, use the **show ip bgp vpnv4** EXEC command or the **show ip route vrf** EXEC command.

In the following example, the **show ip bgp vpnv4** command is entered to display multipaths installed in the VPNv4 RIB:

```
Device# show ip bgp vpnv4 all 10.22.22.0

BGP routing table entry for 10:1:22.22.22.0/24, version 19
Paths:(5 available, best #5)
Multipath:eiBGP
  Advertised to non peer-group peers:
  10.0.0.2 10.0.0.3 10.0.0.4 10.0.0.5
  22
    10.0.0.2 (metric 20) from 10.0.0.4 (10.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.4
  22
    10.0.0.2 (metric 20) from 10.0.0.5 (10.0.0.5)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.5
  22
    10.0.0.2 (metric 20) from 10.0.0.2 (10.0.0.2)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:RT:100:1 0x0:0:0
  22
    10.0.0.2 (metric 20) from 10.0.0.3 (10.0.0.3)
      Origin IGP, metric 0, localpref 100, valid, internal, multipath
      Extended Community:0x0:0:0 RT:100:1 0x0:0:0
      Originator:10.0.0.2, Cluster list:10.0.0.3
  22
    10.1.1.12 from 10.1.1.12 (10.22.22.12)
      Origin IGP, metric 0, localpref 100, valid, external, multipath, best
      Extended Community:RT:100:1
```

In the following example, the **show ip route vrf** command is entered to display multipath routes in the VRF table:

```
Device# show ip route vrf PATH 10.22.22.0

Routing entry for 10.22.22.0/24
  Known via "bgp 1", distance 20, metric 0
  Tag 22, type external
  Last update from 10.1.1.12 01:59:31 ago
  Routing Descriptor Blocks:
  * 10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.4, 01:59:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.5, 01:59:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.2, 01:59:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.0.0.2 (Default-IP-Routing-Table), from 10.0.0.3, 01:59:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
    10.1.1.12, from 10.1.1.12, 01:59:31 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
```

Where to Go Next

For information about advertising the bandwidth of an autonomous system exit link as an extended community, refer to the "BGP Link Bandwidth" module.

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
Comprehensive BGP link bandwidth configuration examples and tasks	"BGP Link Bandwidth" module in the *IP Routing: BGP Configuration Guide*
CEF configuration tasks	"CEF Overview" module in the *IP Switching Cisco Express Forwarding Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 1771	*A Border Gateway Protocol 4 (BGP4)*
RFC 2547	*BGP/MPLS VPNs*
RFC 2858	*Multiprotocol Extensions for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

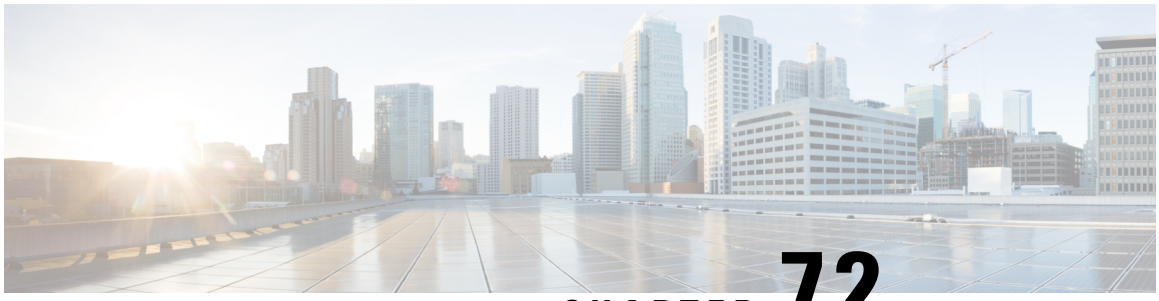
Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 97: Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Feature Name	Releases	Feature Information
BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.



CHAPTER 72

Loadsharing IP Packets over More Than Six Parallel Paths

This document describes the Loadsharing IP Packets over More Than Six Parallel Paths feature, which increases the maximum number of parallel routes that can be installed to the routing table for multipath loadsharing.

Overview of Loadsharing IP Packets over More Than Six Parallel Paths

The Loadsharing IP Packets over More Than Six Parallel Paths feature increases the maximum number of parallel routes that can be installed to the routing table. The maximum number has been increased from six to sixteen for the following commands:

- **maximum-paths**
- **maximum-paths eibgp**
- **maximum-paths ibgp**

The output of the **show ip route summary** command has been updated to display the number of parallel routes supported by the routing table.

The benefits of this feature include the following:

- More flexible configuration of parallel routes in the routing table.
- Ability to configure multipath loadsharing over more links to allow for the configuration of higher-bandwidth aggregation using lower-speed links.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
eBGP multipath load sharing	"BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN" module
iBGP multipath load sharing	"iBGP Multipath Load Sharing" module

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Loadsharing IP Packets over More Than Six Parallel Paths

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 98: Feature Information for Loadsharing IP Packets over More Than Six Parallel Paths

Feature Name	Releases	Feature Information
Loadsharing IP Packets over More Than Six Parallel Paths	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified by this feature: **maximum-paths**, **maximum-paths eibgp**, **maximum-paths ibgp**, **show ip route summary**

CHAPTER 73

BGP Policy Accounting

Border Gateway Protocol (BGP) policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting is enabled on an input interface, and counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic.

Prerequisites

Before using the BGP Policy Accounting feature, you must enable BGP and CEF or dCEF on the router.

Information About BGP Policy Accounting

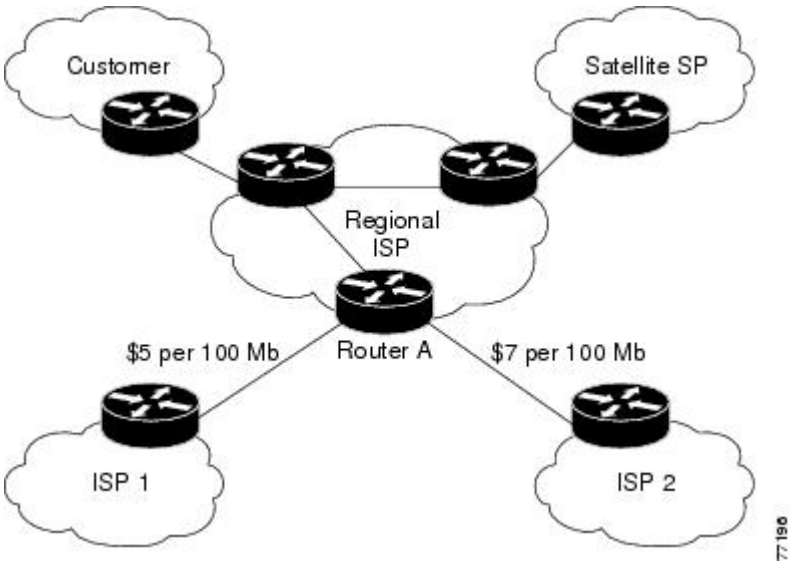
BGP Policy Accounting Overview

Border Gateway Protocol (BGP) policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting is enabled on an input interface, and counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic.

Using the BGP **table-map** command, prefixes added to the routing table are classified by BGP attribute, autonomous system number, or autonomous system path. Packet and byte counters are incremented per input interface. A Cisco IOS policy-based classifier maps the traffic into one of eight possible buckets, representing different traffic classes.

Using BGP policy accounting, you can account for traffic according to the route it traverses. Service providers (SPs) can identify and account for all traffic by customer and bill accordingly. In the figure below, BGP policy accounting can be implemented in Router A to measure packet and byte volumes in autonomous system buckets. Customers are billed appropriately for traffic that is routed from a domestic, international, or satellite source.

Figure 90: Sample Topology for BGP Policy Accounting



BGP policy accounting using autonomous system numbers can be used to improve the design of network circuit peering and transit agreements between Internet service providers (ISPs).

Benefits of BGP Policy Accounting

Account for IP Traffic Differentially

BGP policy accounting classifies IP traffic by autonomous system number, autonomous system path, or community list string, and increments packet and byte counters. Service providers can account for traffic and apply billing, according to the route specific traffic traverses.

Efficient Network Circuit Peering and Transit Agreement Design

Implementing BGP policy accounting on an edge router can highlight potential design improvements for peering and transit agreements.

How to Configure BGP Policy Accounting

Specifying the Match Criteria for BGP Policy Accounting

The first task in configuring BGP policy accounting is to specify the criteria that must be matched. Community lists, autonomous system paths, or autonomous system numbers are examples of BGP attributes that can be specified and subsequently matched using a route map.

To specify the BGP attribute to use for BGP policy accounting and create the match criteria in a route map, use the following commands in global configuration mode:

SUMMARY STEPS

1. Device(config)# **ip community-list** *community-list-number* {**permit** | **deny**} *community-number*
2. Device(config)# **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
3. Device(config-route-map)# **match community-list** *community-list-number* [**exact**]
4. Device(config-route-map)# **set traffic-index** *bucket-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# **ip community-list** *community-list-number* {**permit** | **deny**} *community-number*	Creates a community list for BGP and controls access to it. This step must be repeated for each community to be specified.
Step 2	Device(config)# **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]	Enters route-map configuration mode and defines the conditions for policy routing. The *map-name* argument identifies a route map. The optional **permit** and **deny** keywords work with the match and set criteria to control how the packets are accounted for. The optional *sequence-number* argument indicates the position a new route map is to have in the list of route maps already configured with the same name.
Step 3	Device(config-route-map)# **match community-list** *community-list-number* [**exact**]	Matches a BGP community.
Step 4	Device(config-route-map)# **set traffic-index** *bucket-number*	Indicates where to output packets that pass a match clause of a route map for BGP policy accounting.

Classifying the IP Traffic and Enabling BGP Policy Accounting

After a route map has been defined to specify match criteria, you must configure a way to classify the IP traffic before enabling BGP policy accounting.

Using the **table-map** command, BGP classifies each prefix it adds to the routing table based on the match criteria. When the **bgp-policy accounting** command is configured on an interface, BGP policy accounting is enabled.

To classify the IP traffic and enable BGP policy accounting, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Device(config)# **router bgp** *as-number*
2. Device(config-router)# **table-map** *route-map-name*
3. Device(config-router)# **network** *network-number* [**mask** *network-mask*]
4. Device(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
5. Device(config-router)# **exit**

6. Device(config)# **interface** *interface-type interface-number*
7. Device(config-if)# **no ip directed-broadcast**
8. Device(config-if)# **ip address** *ip-address mask*
9. Device(config-if)# **bgp-policy accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# **router bgp** *as-number*	Configures a BGP routing process and enters router configuration mode for the specified routing process.
Step 2	Device(config-router)# **table-map** *route-map-name*	Classifies BGP prefixes entered in the routing table.
Step 3	Device(config-router)# **network** *network-number* [**mask** *network-mask*]	Specifies a network to be advertised by the BGP routing process.
Step 4	Device(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Specifies a BGP peer by adding an entry to the BGP routing table.
Step 5	Device(config-router)# **exit**	Exits to global configuration mode.
Step 6	Device(config)# **interface** *interface-type interface-number*	Specifies the interface type and number and enters interface configuration mode.
Step 7	Device(config-if)# **no ip directed-broadcast**	Configures the interface to drop directed broadcasts destined for the subnet to which that interface is attached, rather than being broadcast. This is a security issue.
Step 8	Device(config-if)# **ip address** *ip-address mask*	Configures the interface with an IP address.
Step 9	Device(config-if)# **bgp-policy accounting**	Enables BGP policy accounting for the interface.

Verifying BGP Policy Accounting

To verify that BGP policy accounting is operating, perform the following steps:

SUMMARY STEPS

1. Enter the **show ip cef** EXEC command with the **detail** keyword to learn which accounting bucket is assigned to a specified prefix.
2. Enter the **show ip bgp** EXEC command for the same prefix used in Step 1--192.168.5.0-- to learn which community is assigned to this prefix.
3. Enter the **show cef interface policy-statistics** EXEC command to display the per-interface traffic statistics.

DETAILED STEPS

Step 1 Enter the **show ip cef** EXEC command with the **detail** keyword to learn which accounting bucket is assigned to a specified prefix.

In this example, the output is displayed for the prefix 192.168.5.0. It shows that the accounting bucket number 4 (traffic_index 4) is assigned to this prefix.

Example:

```
Device# show ip cef 192.168.5.0 detail

192.168.5.0/24, version 21, cached adjacency to POS7/2
0 packets, 0 bytes, traffic_index 4
  via 10.14.1.1, 0 dependencies, recursive
   next hop 10.14.1.1, POS7/2 via 10.14.1.0/30
   valid cached adjacency
```

Step 2 Enter the **show ip bgp** EXEC command for the same prefix used in Step 1--192.168.5.0-- to learn which community is assigned to this prefix.

In this example, the output is displayed for the prefix 192.168.5.0. It shows that the community of 100:197 is assigned to this prefix.

Example:

```
Device# show ip bgp 192.168.5.0

BGP routing table entry for 192.168.5.0/24, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  100
    10.14.1.1 from 10.14.1.1 (32.32.32.32)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 100:197
```

Step 3 Enter the **show cef interface policy-statistics** EXEC command to display the per-interface traffic statistics.

In this example, the output shows the number of packets and bytes that have been assigned to each accounting bucket:

Example:

```
Device# show cef interface policy-statistics

POS7/0 is up (if_number 8)
Bucket    Packets               Bytes
1          0                      0
2          0                      0
3          50                     5000
4          100                    10000
5          100                    10000
6          10                     1000
7          0                      0
8          0                      0
```

Monitoring and Maintaining BGP Policy Accounting

Command	Purpose
`Device# show cef interface [type number] policy-statistics`	(Optional) Displays detailed CEF policy statistical information for all interfaces.

Command	Purpose
`Device# show ip bgp [network] [network mask] [longer-prefixes]`	(Optional) Displays entries in the BGP routing table.
`Device# show ip cef [network [mask]] [detail]`	(Optional) Displays entries in the Forwarding Information Base (FIB) or FIB summary information.

Configuration Examples for BGP Policy Accounting

Specifying the Match Criteria for BGP Policy Accounting Example

In the following example, BGP communities are specified in community lists, and a route map named set_bucket is configured to match each of the community lists to a specific accounting bucket using the **set traffic-index** command:

```
ip community-list 30 permit 100:190
ip community-list 40 permit 100:198
ip community-list 50 permit 100:197
ip community-list 60 permit 100:296
!
route-map set_bucket permit 10
 match community-list 30
 set traffic-index 2
!
route-map set_bucket permit 20
 match community-list 40
 set traffic-index 3
!
route-map set_bucket permit 30
 match community-list 50
 set traffic-index 4
!
route-map set_bucket permit 40
 match community-list 60
 set traffic-index 5
```

Example: Classifying the IP Traffic and Enabling BGP Policy Accounting

In the following example, BGP policy accounting is enabled on POS interface 7/0 and the **table-map** command is used to modify the bucket number when the IP routing table is updated with routes learned from BGP:

```
router bgp 65000
 table-map set_bucket
 network 10.15.1.0 mask 255.255.255.0
 neighbor 10.14.1.1 remote-as 65100
!
ip classless
ip bgp-community new-format
!
interface POS7/0
```

```
ip address 10.15.1.2 255.255.255.0
no ip directed-broadcast
bgp-policy accounting
no keepalive
crc 32
clock source internal
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Cisco Express Forwarding (CEF) and distributed CEF (dCEF) commands	Cisco IOS IP Switching Command Reference
Cisco Express Forwarding (CEF) and distributed CEF (dCEF) configuration information	"CEF Overview" module of the *Cisco IOS Switching Services Configuration Guide*

MIBs

MIB	MIBs Link
• CISCO-BGP-POLICY-ACCOUNTING-MIB **Note** CISCO-BGP-POLICY-ACCOUNTING-MIB is only available in the Cisco IOS Release 12.0(9)S, 12.0(17)ST, and later releases. This MIB is not available on any mainline and T-train release.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

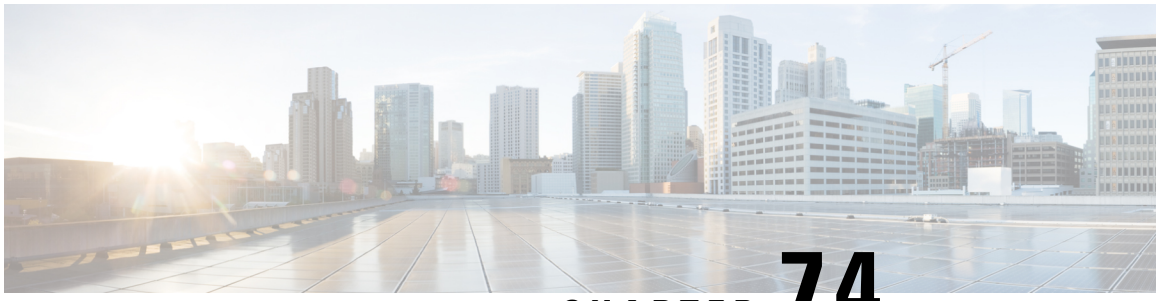
Feature Information for BGP Policy Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 99: Feature Information for BGP Policy Accounting

Feature Name	Releases	Feature Information
BGP Policy Accounting	12.0(9)S 12.0(17)ST 12.2(13)T 15.0(1)S 12.2(50)SY Cisco IOS XE Release 3.8S	Border Gateway Protocol (BGP) policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting is enabled on an input interface, and counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic. The following commands were introduced or modified: • **bgp-policy** • **set traffic-index** • **show cef interface policy-statistics** • **show ip bgp** • **show ip cef**



CHAPTER 74

BGP Policy Accounting Output Interface Accounting

Border Gateway Protocol (BGP) policy accounting (PA) measures and classifies IP traffic that is sent to, or received from, different peers. Policy accounting was previously available on an input interface only. The BGP Policy Accounting Output Interface Accounting feature introduces several extensions to enable BGP PA on an output interface and to include accounting based on a source address for both input and output traffic on an interface. Counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic.

Prerequisites for BGP PA Output Interface Accounting

Before using the BGP Policy Accounting Output Interface Accounting feature, you must enable BGP and Cisco Express Forwarding or distributed CEF on the router.

Information About BGP PA Output Interface Accounting

BGP PA Output Interface Accounting

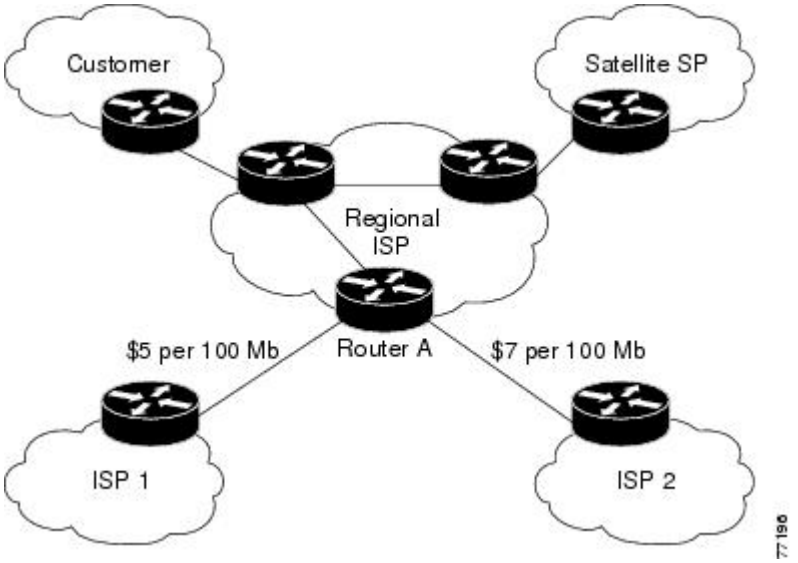
Policy accounting using BGP measures and classifies IP traffic that is sent to, or received from, different peers. Originally, BGP PA was available on an input interface only. BGP PA output interface accounting introduces several extensions to enable BGP PA on an output interface and to include accounting based on a source address for both input and output traffic on an interface. Counters based on parameters such as community list, autonomous system number, or autonomous system path are assigned to identify the IP traffic.

Using the BGP **table-map** command, prefixes added to the routing table are classified by BGP attribute, autonomous system number, or autonomous system path. Packet and byte counters are incremented per input

or output interface. A Cisco policy-based classifier maps the traffic into one of eight possible buckets that represent different traffic classes.

Using BGP PA, you can account for traffic according to its origin or the route it traverses. Service providers (SPs) can identify and account for all traffic by customer and can bill accordingly. In the figure below, BGP PA can be implemented in Router A to measure packet and byte volumes in autonomous system buckets. Customers are billed appropriately for traffic that is routed from a domestic, international, or satellite source.

Figure 91: Sample Topology for BGP Policy Accounting



BGP policy accounting using autonomous system numbers can be used to improve the design of network circuit peering and transit agreements between Internet service providers (ISPs).

Benefits of BGP PA Output Interface Accounting

Accounting for IP Traffic Differentially

BGP policy accounting classifies IP traffic by autonomous system number, autonomous system path, or community list string, and increments packet and byte counters. Policy accounting can also be based on the source address. Service providers can account for traffic and apply billing according to the origin of the traffic or the route that specific traffic traverses.

Efficient Network Circuit Peering and Transit Agreement Design

Implementing BGP policy accounting on an edge router can highlight potential design improvements for peering and transit agreements.

How to Configure BGP PA Output Interface Accounting

Specifying the Match Criteria for BGP PA

The first task in configuring BGP PA is to specify the criteria that must be matched. Community lists, autonomous system paths, or autonomous system numbers are examples of BGP attributes that can be specified and subsequently matched using a route map. Perform this task to specify the BGP attribute to use for BGP PA and to create the match criteria in a route map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip community-list** {*standard-list-number* | *expanded-list-number* [*regular-expression*] | {**standard** | **expanded**} *community-list-name*} {**permit** | **deny**} {*community-number* | *regular-expression*}
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
5. **match community-list** *community-list-number* [**exact**]
6. **set traffic-index** *bucket-number*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip community-list** {*standard-list-number* | *expanded-list-number* [*regular-expression*] | {**standard** | **expanded**} *community-list-name*} {**permit** | **deny**} {*community-number* | *regular-expression*} **Example:** `Device(config)# ip community-list 30 permit 100:190`	Creates a community list for BGP and controls access to it. • Repeat this step for each community to be specified.
Step 4	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map set_bucket permit 10`	Enters route-map configuration mode and defines the conditions for policy routing. • The *map-name* argument identifies a route map.

	Command or Action	Purpose
		• The optional **permit** and **deny** keywords work with the match and set criteria to control how the packets are accounted for. • The optional *sequence-number* argument indicates the position that a new route map is to have in the list of route maps already configured with the same name.
Step 5	**match community-list** *community-list-number* [**exact**] **Example:** `Router(config-route-map)# match community-list 30`	Matches a BGP community.
Step 6	**set traffic-index** *bucket-number* **Example:** `Device(config-route-map)# set traffic-index 2`	Indicates where to output packets that pass a match clause of a route map for BGP policy accounting.
Step 7	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.

Classifying the IP Traffic and Enabling BGP PA

After a route map has been defined to specify match criteria, you must configure a way to classify the IP traffic before enabling BGP policy accounting.

Using the **table-map** command, BGP classifies each prefix that it adds to the routing table according to the match criteria. When the **bgp-policy accounting** command is configured on an interface, BGP policy accounting is enabled.

Perform this task to classify the IP traffic and enable BGP policy accounting.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **table-map** *route-map-name*
5. **network** *network-number* [**mask** *network-mask*]
6. **neighbor** *ip-address* **remote-as** *as-number*
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **bgp-policy accounting** [**input** | **output**] [**source**]
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode for the specified routing process. • The *as-number* argument identifies a BGP autonomous system number.
Step 4	**table-map** *route-map-name* **Example:** `Device(config-router)# table-map set_bucket`	Classifies BGP prefixes entered in the routing table.
Step 5	**network** *network-number* [**mask** *network-mask*] **Example:** `Device(config-router)# network 10.15.1.0 mask 255.255.255.0`	Specifies a network to be advertised by the BGP routing process.
Step 6	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 10.14.1.1 remote-as 65100`	Specifies a BGP peer by adding an entry to the BGP routing table.
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface POS 7/0`	Specifies the interface type and number and enters interface configuration mode. • The *type* argument identifies the type of interface. • The *number* argument identifies the slot and port numbers of the interface. The space between the interface type and number is optional.

	Command or Action	Purpose
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip-address 10.15.1.2 255.255.255.0`	Configures the interface with an IP address.
Step 10	**bgp-policy accounting** [**input** | **output**] [**source**] **Example:** `Device(config-if)# bgp-policy accounting input source`	Enables BGP policy accounting for the interface. • Use the optional **input** or **output** keyword to account for traffic either entering or leaving the router. By default, BGP policy accounting is based on traffic entering the router. • Use the optional **source** keyword to account for traffic based on source address.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Verifying BGP Policy Accounting

Perform this task to verify that BGP policy accounting is operating.

SUMMARY STEPS

1. **show ip cef** [*network* [*mask*]] [**detail**]
2. **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]
3. **show cef interface** [*type number*] **policy-statistics** [**input** | **output**]
4. **show cef interface** [*type number*] [**statistics**] [**detail**]

DETAILED STEPS

Step 1 **show ip cef** [*network* [*mask*]] [**detail**]

Enter the **show ip cef** command with the **detail** keyword to learn which accounting bucket is assigned to a specified prefix.

In this example, the output is displayed for the prefix 192.168.5.0. It shows that accounting bucket number 4 (traffic_index 4) is assigned to this prefix.

Example:

```
Device# show ip cef 192.168.5.0 detail

192.168.5.0/24, version 21, cached adjacency to POS7/2
0 packets, 0 bytes, traffic_index 4
  via 10.14.1.1, 0 dependencies, recursive
```

```
   next hop 10.14.1.1, POS7/2 via 10.14.1.0/30
   valid cached adjacency
```

Step 2 **show ip bgp** [*network*] [*network-mask*] [**longer-prefixes**]

Enter the **show ip bgp** command for the same prefix used in Step 1--192.168.5.0--to learn which community is assigned to this prefix.

In this example, the output is displayed for the prefix 192.168.5.0. It shows that the community of 100:197 is assigned to this prefix.

Example:

```
Device# show ip bgp 192.168.5.0

BGP routing table entry for 192.168.5.0/24, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  100
    10.14.1.1 from 10.14.1.1 (32.32.32.32)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: 100:197
```

Step 3 **show cef interface** [*type number*] **policy-statistics** [**input** | **output**]

Displays the per-interface traffic statistics.

In this example, the output shows the number of packets and bytes that have been assigned to each accounting bucket:

Example:

```
Device# show cef interface policy-statistics input

FastEthernet1/0/0 is up (if_number 6)
  Corresponding hwidb fast_if_number 6
  Corresponding hwidb firstsw->if_number 6
  BGP based Policy accounting on input is enabled
 Index         Packets          Bytes
     1            9999         999900
     2               0              0
     3               0              0
     4               0              0
     5               0              0
     6               0              0
     7               0              0
     8               0              0
     9               0              0
    10               0              0
    11               0              0
    12               0              0
    13               0              0
    14               0              0
    15               0              0
    16               0              0
    17               0              0
    18               0              0
    19               0              0
    20               0              0
    21               0              0
    22               0              0
    23               0              0
    24               0              0
    25               0              0
```

```
  26              0              0
  27              0              0
  28              0              0
  29              0              0
  30              0              0
  31              0              0
  32              0              0
  33              0              0
  34           1234         123400
  35              0              0
  36              0              0
  37              0              0
  38              0              0
  39              0              0
  40              0              0
  41              0              0
  42              0              0
  43              0              0
  44              0              0
  45           1000         100000
  46              0              0
  47              0              0
  48              0              0
  49              0              0
  50              0              0
  51              0              0
  52              0              0
  53              0              0
  54           5123        1198782
  55              0              0
  56              0              0
  57              0              0
  58              0              0
  59              0              0
  60              0              0
  61              0              0
  62              0              0
  63              0              0
  64              0              0
```

Step 4 **show cef interface** [*type number*] [**statistics**] [**detail**]

Displays the state of BGP policy accounting on a specified interface.

In this example, the output shows that BGP policy accounting has been configured to be based on input traffic at Fast Ethernet interface 1/0/0:

Example:

```
Device# show cef interface Fast Ethernet 1/0/0

FastEthernet1/0/0 is up (if_number 6)
  Corresponding hwidb fast_if_number 6
  Corresponding hwidb firstsw->if_number 6
  Internet address is 10.1.1.1/24
  ICMP redirects are always sent
  Per packet load-sharing is disabled
  IP unicast RPF check is disabled
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  BGP based policy accounting on input is enabled
  BGP based policy accounting on output is disabled
  Hardware idb is FastEthernet1/0/0 (6)
```

```
Software idb is FastEthernet1/0/0 (6)
Fast switching type 1, interface type 18
IP Distributed CEF switching enabled
IP Feature Fast switching turbo vector
IP Feature CEF switching turbo vector
Input fast flags 0x100, Output fast flags 0x0, Flags 0x0
ifindex 7(7)
Slot 1 Slot unit 0 VC -1
Transmit limit accumulator 0xE8001A82 (0xE8001A82)
IP MTU 1500
```

Configuration Examples for BGP PA Output Interface Accounting

Specifying the Match Criteria for BGP Policy Accounting Example

In the following example, BGP communities are specified in community lists, and a route map named set_bucket is configured to match each of the community lists to a specific accounting bucket using the **set traffic-index** command:

```
ip community-list 30 permit 100:190
ip community-list 40 permit 100:198
ip community-list 50 permit 100:197
ip community-list 60 permit 100:296
!
route-map set_bucket permit 10
 match community-list 30
 set traffic-index 2
!
route-map set_bucket permit 20
 match community-list 40
 set traffic-index 3
!
route-map set_bucket permit 30
 match community-list 50
 set traffic-index 4
!
route-map set_bucket permit 40
 match community-list 60
 set traffic-index 5
```

Classifying the IP Traffic and Enabling BGP Policy Accounting Example

In the following example, BGP policy accounting is enabled on POS interface 2/0/0. The policy accounting criteria is based on the source address of the input traffic, and the **table-map** command is used to modify the bucket number when the IP routing table is updated with routes learned from BGP.

```
router bgp 65000
 table-map set_bucket
 network 10.15.1.0 mask 255.255.255.0
 neighbor 10.14.1.1 remote-as 65100
!
ip classless
ip bgp-community new-format
```

```
!
interface POS2/0/0
 ip address 10.15.1.2 255.255.255.0
 bgp-policy accounting input source
 no keepalive
 crc 32
 clock source internal
```

Additional References

The following sections provide references related to the BGP policy accounting output interface accounting feature.

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*
Switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IP Switching Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
CISCO-BGP-POLICY-ACCOUNTING-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for BGP Policy Accounting Output Interface Accounting

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 100: Feature Information for BGP Policy Accounting Output Interface Accounting

Feature Name	Releases	Feature Information
BGP Policy Accounting	Cisco IOS XE Release 2.1	BGP policy accounting measures and classifies IP traffic that is sent to, or received from, different peers. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
BGP Policy Accounting Output Interface Accounting	Cisco IOS XE Release 2.1	This feature introduces several extensions to enable BGP PA on an output interface and to include accounting based on a source address for both input and output traffic on an interface. This feature was introduced on the Cisco ASR 1000 Series Routers. The following commands were introduced or modified for this feature: **bgp-policy**, **set traffic-index**, **show cef interface**, **show cef interface policy-statistics**
SNMP Support for BGP Policy Accounting	Cisco IOS XE Release 2.1	The CISCO-BGP-POLICY-ACCOUNTING-MIB was introduced. This feature was introduced on the Cisco ASR 1000 Series Routers.

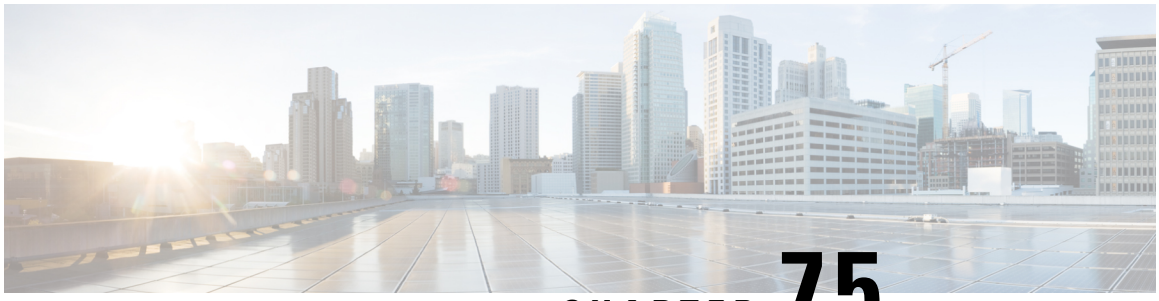
Glossary

AS --autonomous system. An IP term to describe a routing domain that has its own independent routing policy and is administered by a single authority.

BGP --Border Gateway Protocol. Interdomain routing protocol that exchanges reachability information with other BGP systems.

CEF --Cisco Express Forwarding.

dCEF --distributed Cisco Express Forwarding.



CHAPTER 75

BGP Cost Community

The BGP Cost Community feature introduces the cost extended community attribute. The cost community is a non-transitive extended community attribute that is passed to internal BGP (iBGP) and confederation peers but not to external BGP (eBGP) peers. The cost community feature allows you to customize the local route preference and influence the best path selection process by assigning cost values to specific routes.

In Cisco IOS Release 12.0(27)S, 12.3(8)T, 12.2(25)S, and later releases, support was introduced for mixed EIGRP MPLS VPN network topologies that contain VPN and backdoor links.

Prerequisites for the BGP Cost Community Feature

This document assumes that BGP is configured in your network and that peering has been established.

Restrictions for the BGP Cost Community Feature

- The BGP Cost Community feature can be configured only within an autonomous system or confederation. The cost community is a non-transitive extended community that is passed to iBGP and confederation peers only and is not passed to eBGP peers.
- The BGP Cost Community feature must be supported on all routers in the autonomous system or confederation before cost community filtering is configured. The cost community should be applied consistently throughout the local autonomous system or confederation to avoid potential routing loops.
- Multiple cost community set clauses may be configured with the **set extcommunity cost** command in a single route map block or sequence. However, each set clause must be configured with a different ID value (0-255) for each point of insertion (POI). The ID value determines preference when all other attributes are equal. The lowest ID value is preferred.

Information About the BGP Cost Community Feature

BGP Cost Community Overview

The cost community is a nontransitive, extended community attribute that is passed to iBGP and confederation peers, but not to eBGP peers. The configuration of the BGP Cost Community feature allows you to customize the BGP best path selection process for a local autonomous system or confederation.

The cost community attribute is applied to internal routes by configuring the **set extcommunity cost** command in a route map. The cost community set clause is configured with a cost community ID number (0-255) and cost number (0-4294967295). The cost community ID number determines the preference for the path selection process. The path with the lowest cost community ID number is preferred.

Paths that are not specifically configured with the cost community attribute are assigned a default cost number value of 2147483647 (The midpoint between 0 and 4294967295) and evaluated by the best path selection process accordingly. In the case where two paths have been configured with the same cost community ID number, the path selection process will then prefer the path with the lowest cost number. The cost extended community attribute is propagated to iBGP peers when extended community exchange is enabled with the **neighbor send-community** command.

The following commands can be used to apply a route map that is configured with the cost community set clause:

- **aggregate-address**
- **neighbor default-originate route-map {in | out}**
- **neighbor route-map**
- **network route-map**
- **redistribute route-map**

How the BGP Cost Community Influences the Best Path Selection Process

The cost community attribute influences the BGP best path selection process at the point of insertion (POI). By default, the POI follows the IGP metric comparison. When BGP receives multiple paths to the same destination, it uses the best path selection process to determine which path is the best path. BGP automatically makes the decision and installs the best path into the routing table. The POI allows you to assign a preference to o a specific path when multiple equal cost paths are available. If the POI is not valid for local best path selection, the cost community attribute is silently ignored.

Multiple paths can be configured with the cost community attribute for the same POI. The path with the lowest cost community ID is considered first. In other words, all of the cost community paths for a specific POI are considered, starting with the one with the lowest cost community. Paths that do not contain the cost community (for the POI and community ID being evaluated) are assigned the default community cost value (2147483647). If the cost community values are equal, then cost community comparison proceeds to the next lowest community ID for this POI.

Note Paths that are not configured with the cost community attribute are considered by the best path selection process to have the default cost-value (half of the maximum value [4294967295] or 2147483647).

Applying the cost community attribute at the POI allows you to assign a value to a path originated or learned by a peer in any part of the local autonomous system or confederation. The cost community can be used as a "tie breaker" during the best path selection process. Multiple instances of the cost community can be configured for separate equal cost paths within the same autonomous system or confederation. For example, a lower cost community value can be applied to a specific exit path in a network with multiple equal cost exits points, and the specific exit path will be preferred by the BGP best path selection process. See the scenario described in the "Influencing Route Preference in a Multi-Exit IGP Network" section.

Cost Community Support for Aggregate Routes and Multipaths

Aggregate routes and multipaths are supported by the BGP Cost Community feature. The cost community attribute can be applied to either type of route. The cost community attribute is passed to the aggregate or multipath route from component routes that carry the cost community attribute. Only unique IDs are passed, and only the highest cost of any individual component route will be applied to the aggregate on a per-ID basis. If multiple component routes contain the same ID, the highest configured cost is applied to the route. For example, the following two component routes are configured with the cost community attribute via an inbound route map:

- 10.0.0.1 (POI=IGP, ID=1, Cost=100)
- 192.168.0.1 (POI=IGP, ID=1, Cost=200)

If these component routes are aggregated or configured as a multipath, the cost value 200 (POI=IGP, ID=1, Cost=200) will be advertised because it is the highest cost.

If one or more component routes does not carry the cost community attribute or if the component routes are configured with different IDs, then the default value (2147483647) will be advertised for the aggregate or multipath route. For example, the following three component routes are configured with the cost community attribute via an inbound route map. However, the component routes are configured with two different IDs.

- 10.0.0.1 (POI=IGP, ID=1, Cost=100)
- 172.16.0.1 (POI=IGP, ID=2, Cost=100)
- 192.168.0.1 (POI=IGP, ID=1, Cost=200)

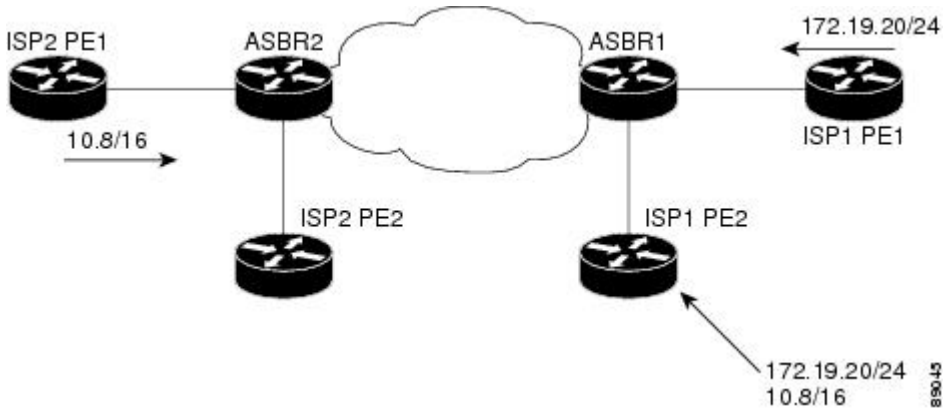
The single advertised path will include the aggregated cost communities as follows:

- {POI=IGP, ID=1, Cost=2147483647} {POI=IGP, ID=2, Cost=2147483647}

Influencing Route Preference in a Multi-Exit IGP Network

The figure below shows an Interior Gateway Protocol (IGP) network with two autonomous system boundary routers (ASBRs) on the edge. Each ASBR has an equal cost path to network 10.8/16.

Figure 92: Multi-Exit Point IGP Network



Both paths are considered to be equal by BGP. If multipath loadsharing is configured, both paths will be installed to the routing table and will be used to load balance traffic. If multipath load balancing is not configured, then BGP will select the path that was learned first as the best path and install this path to the routing table. This behavior may not be desirable under some conditions. For example, the path is learned from ISP1 PE2 first, but the link between ISP1 PE2 and ASBR1 is a low-speed link.

The configuration of the cost community attribute can be used to influence the BGP best path selection process by applying a lower cost community value to the path learned by ASBR2. For example, the following configuration is applied to ASBR2.

```
route-map ISP2_PE1 permit 10
 set extcommunity cost 1 1
 match ip address 13
!
ip access-list 13 permit 10.8.0.0 0.0.255.255
```

The above route map applies a cost community number value of 1 to the 10.8.0.0 route. By default, the path learned from ASBR1 will be assigned a cost community value of 2147483647. Because the path learned from ASBR2 has lower cost community value, this path will be preferred.

BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links

Before EIGRP Site of Origin (SoO) BGP Cost Community support was introduced, BGP preferred locally sourced routes over routes learned from BGP peers. Back door links in an EIGRP MPLS VPN topology will be preferred by BGP if the back door link is learned first. (A back door link, or a route, is a connection that is configured outside of the VPN between a remote and main site. For example, a WAN leased line that connects a remote site to the corporate network).

The "pre-bestpath" point of insertion (POI) was introduced in the BGP Cost Community feature to support mixed EIGRP VPN network topologies that contain VPN and backdoor links. This POI is applied automatically to EIGRP routes that are redistributed into BGP. The "pre-best path" POI carries the EIGRP route type and metric. This POI influences the best path calculation process by influencing BGP to consider this POI before any other comparison step. No configuration is required. This feature is enabled automatically for EIGRP VPN sites when Cisco IOS Release 12.0(27)S is installed to a PE, CE, or back door router.

For information about configuring EIGRP MPLS VPNs, refer to the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge document in Cisco IOS Release 12.0(27)S.

For more information about the EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature, refer to the EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature documentation in Cisco IOS Release 12.0(27)S.

How to Configure the BGP Cost Community Feature

Configuring the BGP Cost Community

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **ipv6** [**multicast** | **unicast**] | **vpnv4** [**unicast**]
6. **neighbor** *ip-address* **route-map** *map-name* {**in** | **out**}
7. **exit**
8. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
9. **set extcommunity cost** [**igp**] *community-id cost-value*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.0.0.1 remote-as 101`	Establishes peering with the specified neighbor or peer-group.

	Command or Action	Purpose
Step 5	**address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **ipv6** [**multicast** | **unicast**] | **vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family ipv4`	Places the router in address family configuration mode.
Step 6	**neighbor** *ip-address* **route-map** *map-name* {**in** | **out**} **Example:** `Device(config-router)# neighbor 10.0.0.1 route-map MAP-NAME in`	Applies an incoming or outgoing route map for the specified neighbor or peer-group.
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 8	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map MAP-NAME permit 10`	Enters route map configuration mode to create or configure a route map.
Step 9	**set extcommunity cost** [**igp**] *community-id cost-value* **Example:** `Device(config-route-map)# set extcommunity cost 1 100`	Creates a set clause to apply the cost community attribute. • Multiple cost community set clauses can be configured in each route map block or sequence. Each cost community set clause must have a different ID (0-255). The cost community set clause with the lowest *cost-value* is preferred by the best path selection process when all other attributes are equal. • Paths that are not configured with the cost community attribute will be assigned the default *cost-value*, which is half of the maximum value (4294967295) or 2147483647.
Step 10	**end** **Example:** `Device(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.

Verifying the Configuration of the BGP Cost Community

BGP cost community configuration can be verified locally or for a specific neighbor. To verify the local configuration cost community, use the **show route-map** or **show running-config** command.

To verify that a specific neighbor carries the cost community, use the **show ip bgp** *ip-address* command. The output from these commands displays the POI (IGP is the default POI), the configured ID, and configured cost. For large cost community values, the output from these commands will also show, with + and - values, the difference between the configured cost and the default cost. See "Example: BGP Cost Community Verification" section for sample output.

Troubleshooting Tips

The **bgp bestpath cost-community ignore** command can be used to disable the evaluation of the cost community attribute to help isolate problems and troubleshoot issues that relate to BGP best path selection.

The **debug ip bgp updates** command can be used to print BGP update messages. The cost community extended community attribute will be displayed in the output of this command when received from a neighbor. A message will also be displayed if a non-transitive extended community if received from an external peer.

Configuration Examples for the BGP Cost Community Feature

Example: BGP Cost Community Configuration

The following example applies the cost community ID of 1 and cost community value of 100 to routes that are permitted by the route map. This configuration will cause the best path selection process to prefer this route over other equal-cost paths that were not permitted by this route map sequence.

```
Device(config)# router bgp 50000
Device(config-router)# neighbor 10.0.0.1 remote-as 50000
Device(config-router)# neighbor 10.0.0.1 update-source Loopback 0
Device(config-router)# address-family ipv4
Device(config-router-af)# neighbor 10.0.0.1 activate
Device(config-router-af)# neighbor 10.0.0.1 route-map COST1 in
Device(config-router-af)# neighbor 10.0.0.1 send-community both
Router(config-router-af)# exit
Device(config)# route-map COST1 permit 10
Device(config-route-map)# match ip-address 1
Device(config-route-map)# set extcommunity cost 1 100
```

Example: BGP Cost Community Verification

BGP cost community configuration can be verified locally or for a specific neighbor. To verify the local configuration cost community, use the **show route-map** or **show running-config** command. To verify that a specific neighbor carries the cost community, use the **show ip bgp** *ip-address* command.

The output of the **show route-map** command will display locally configured route-maps, match, set, continue clauses, and the status and configuration of the cost community attribute. The following sample output is similar to the output that will be displayed:

```
Device# show route-map

route-map COST1, permit, sequence 10
  Match clauses:
    as-path (as-path filter): 1
  Set clauses:
    extended community Cost:igp:1:100
```

```
  Policy routing matches: 0 packets, 0 bytes
route-map COST1, permit, sequence 20
  Match clauses:
    ip next-hop (access-lists): 2
  Set clauses:
    extended community Cost:igp:2:200
  Policy routing matches: 0 packets, 0 bytes
route-map COST1, permit, sequence 30
  Match clauses:
    interface FastEthernet0/0
    extcommunity (extcommunity-list filter):300
  Set clauses:
    extended community Cost:igp:3:300
  Policy routing matches: 0 packets, 0 bytes
```

The following sample output shows locally configured routes with large cost community values:

```
Device# show route-map

route-map set-cost, permit, sequence 10
  Match clauses:
  Set clauses:
    extended community RT:1:1 RT:2:2 RT:3:3 RT:4:4 RT:5:5 RT:6:6 RT:7:7
      RT:100:100 RT:200:200 RT:300:300 RT:400:400 RT:500:500 RT:600:600
      RT:700:700 additive
    extended community Cost:igp:1:4294967295 (default+2147483648)
      Cost:igp:2:200 Cost:igp:3:300 Cost:igp:4:400
      Cost:igp:5:2147483648 (default+1) Cost:igp:6:2147484648 (default+1001)
      Cost:igp:7:2147284648 (default-198999)
  Policy routing matches: 0 packets, 0 bytes
```

The output of the **show running config** command will display match, set, and continue clauses that are configured within a route-map. The following sample output is filtered to show only the relevant part of the running configuration:

```
Device# show running-config | begin route-map

route-map COST1 permit 20
 match ip next-hop 2
 set extcommunity cost igp 2 200
!
route-map COST1 permit 30
 match interface FastEthernet0/0
 match extcommunity 300
 set extcommunity cost igp 3 300
.
.
.
```

The output of the **show ip bgp** *ip-address* command can be used to verify if a specific neighbor carries a path that is configured with the cost community attribute. The cost community attribute information is displayed in the “Extended Community” field. The POI, the cost community ID, and the cost community number value are displayed. The following sample output shows that neighbor 172.16.1.2 carries a cost community with an ID of 1 and a cost of 100:

```
Device# show ip bgp 10.0.0.0

BGP routing table entry for 10.0.0.0/8, version 2
Paths: (1 available, best #1)
  Not advertised to any peer
  2 2 2
```

```
172.16.1.2 from 172.16.1.2 (172.16.1.2)
  Origin IGP, metric 0, localpref 100, valid, external, best
  Extended Community: Cost:igp:1:100
```

If the specified neighbor is configured with the default cost community number value or if the default value is assigned automatically for cost community evaluation, “default” with + and - values will be displayed after the cost community number value in the output.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature	“EIGRP MPLS VPN PE-CE Site of Origin (SoO)” module in the *IP Routing: EIGRP Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
draft-retana-bgp-custom-decision-00.txt	BGP Custom Decision Process

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

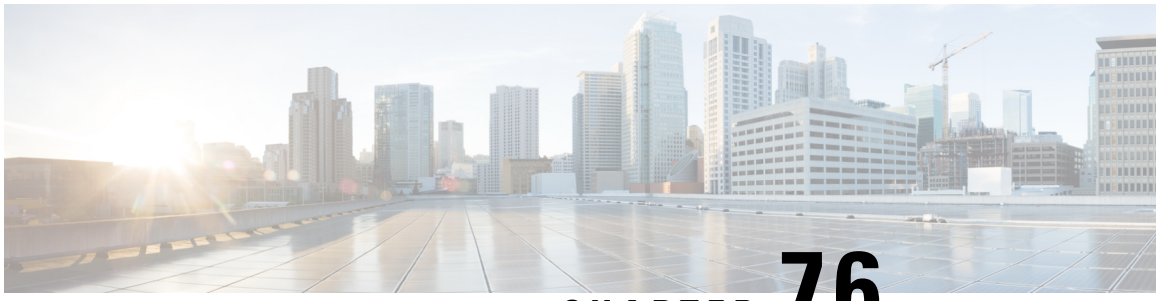
Feature Information for BGP Cost Community

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 101: Feature Information for BGP Cost Community

Feature Name	Releases	Feature Information
BGP Cost Community	12.0(24)S 12.3(2)T 12.2(18)S 12.2(27)SBC 15.0(1)S	The BGP Cost Community feature introduces the cost extended community attribute. The cost community is a non-transitive extended community attribute that is passed to internal BGP (iBGP) and confederation peers but not to external BGP (eBGP) peers. The cost community feature allows you to customize the local route preference and influence the best path selection process by assigning cost values to specific routes. The following commands were introduced or modified: **bgp bestpath cost-community ignore**, **debug ip bgp updates**, and **set extcommunity cost**.
BGP Cost Community Support for EIGRP MPLS VPN PE-CE with Backdoor Links	12.0(27)S 12.3(8)T 12.2(25)S	Back door links in an EIGRP MPLS VPN topology will be preferred by BGP if the back door link is learned first. The "pre-bestpath" point of insertion (POI) was introduced in the BGP Cost Community feature to support mixed EIGRP VPN network topologies that contain VPN and backdoor links. This POI is applied automatically to EIGRP routes that are redistributed into BGP and the POI influences the best path calculation process by influencing BGP to consider this POI before any other comparison step. No configuration is required. This feature is enabled automatically for EIGRP VPN sites when Cisco IOS Release 12.0(27)S, 12.3(8)T, 12,2(25)S or later releases, is installed to a PE, CE, or back door router. No commands were introduced or modified.



CHAPTER 76

BGP Support for IP Prefix Import from Global Table into a VRF Table

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding (VRF) instance table using an import route map.

Prerequisites for BGP Support for IP Prefix Import from Global Table into a VRF Table

- Border Gateway Protocol (BGP) peering sessions are established.
- CEF or dCEF (for distributed platforms) is enabled on all participating routers.

Restrictions for BGP Support for IP Prefix Import from Global Table into a VRF Table

- Only IPv4 unicast and multicast prefixes can be imported into a VRF with this feature.
- IPv4 prefixes imported into a VRF using this feature cannot be imported into a VPNv4 VRF.
- The global prefixes should be in the BGP table, so that this feature can import them into the BGP VRF table.

- IPv4 prefixes imported into a VRF using this feature cannot be imported into a second VPNv4 VRF.

Information About BGP Support for IP Prefix Import from Global Table into a VRF Table

Importing IPv4 Prefixes into a VRF

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding instance (VRF) table using an import route map. This feature extends the functionality of VRF import-map configuration to allow IPv4 prefixes to be imported into a VRF based on a standard community. Both IPv4 unicast and multicast prefixes are supported. No Multiprotocol Label Switching (MPLS) or route target (import/export) configuration is required.

IP prefixes are defined as match criteria for the import map through standard Cisco filtering mechanisms. For example, an IP access-list, an IP prefix-list, or an IP as-path filter is created to define an IP prefix or IP prefix range, and then the prefix or prefixes are processed through a match clause in a route map. Prefixes that pass through the route map are imported into the specified VRF per the import map configuration.

Black Hole Routing

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be configured to support Black Hole Routing (BHR). BHR is a method that allows the administrator to block undesirable traffic, such as traffic from illegal sources or traffic generated by a Denial of Service (DoS) attack, by dynamically routing the traffic to a dead interface or to a host designed to collect information for investigation, mitigating the impact of the attack on the network. Prefixes are looked up, and packets that come from unauthorized sources are rendered as null routes by the ASIC at line rate.

Classifying Global Traffic

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be used to classify global IP traffic based on physical location or class of service. Traffic is classified based on administration policy and then imported into different VRFs. On a college campus, for example, network traffic could be divided into an academic network and residence network traffic, a student network and faculty network, or a dedicated network for multicast traffic. After the traffic is divided along administration policy, routing decisions can be configured with the MPLS VPN--VRF Selection Using Policy Based Routing feature or the MPLS VPN--VRF Selection Based on Source IP Address feature.

Unicast Reverse Path Forwarding

Unicast Reverse Path Forwarding (Unicast RPF) can be optionally configured with the BGP Support for IP Prefix Import from Global Table into a VRF Table feature. Unicast RPF is used to verify that the source address is in the Forwarding Information Base (FIB). The **ip verify unicast vrf** command is configured in interface configuration mode and is enabled for each VRF. This command has **permit** and **deny**keywords that are used to determine if the traffic is forwarded or dropped after Unicast RPF verification.

How to Import IP Prefixes from Global Table into a VRF Table

Defining IPv4 IP Prefixes to Import

IPv4 unicast or multicast prefixes are defined as match criteria for the import route map using standard Cisco filtering mechanisms. This task uses an IP access-list and an IP prefix-list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**]
4. **ip prefix-list** *prefix-list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] [**log**] **Example:** `Device(config)# access-list 50 permit 10.1.1.0 0.0.0.255`	Creates an access list and defines a range of IP prefixes to import into the VRF table. • The example creates a standard access list numbered 50. This filter will permit traffic from any host with an IP address in the 10.1.1.0/24 subnet.
Step 4	**ip prefix-list** *prefix-list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Device(config)# ip prefix-list COLORADO permit 10.24.240.0/22`	Creates a prefix list and defines a range of IP prefixes to import into the VRF table. • The example creates an IP prefix list named COLORADO. This filter will permit traffic from any host with an IP address in the 10.24.240.0/22 subnet.

Creating the VRF and the Import Route Map

The IP prefixes that are defined for import are then processed through a match clause in a route map. IP prefixes that pass through the route map are imported into the VRF. A maximum of 5 VRFs per router can be configured to import IPv4 prefixes from the global routing table. By default, a maximum of 1000 prefixes per VRF can be imported. You can change the limit to be from 1 to 2,147,483,647 prefixes for each VRF. We recommend that you use caution if you increase the prefix import limit above 1000. Configuring the router to import too many prefixes can interrupt normal router operation.

No MPLS or route target (import/export) configuration is required.

Import actions are triggered when a new routing update is received or when routes are withdrawn. During the initial BGP update period, the import action is postponed to allow BGP to convergence more quickly. Once BGP converges, incremental BGP updates are evaluated immediately and qualified prefixes are imported as they are received.

The following syslog message is introduced by the BGP Support for IP Prefix Import from Global Table into a VRF Table feature. It will be displayed when more prefixes are available for import than the user-defined limit:

```
00:00:33: %BGP-3-AFIMPORT_EXCEED: IPv4 Multicast prefixes imported to multicast vrf exceed
 the limit 2
```

You can either increase the prefix limit or fine-tune the import route map filter to reduce the number of candidate routes.

Note

- Only IPv4 unicast and multicast prefixes can be imported into a VRF with this feature.
- A maximum of five VRF instances per router can be created to import IPv4 prefixes from the global routing table.
- IPv4 prefixes imported into a VRF using this feature cannot be imported into a VPNv4 VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **import ipv4** {**unicast** | **multicast**} [*prefix-limit*] **map** *route-map*
6. **exit**
7. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
8. **match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] | **prefix-list** *prefix-list-name* [*prefix-list-name*]}
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf GREEN`	Creates a VRF routing table and specifies the VRF name (or tag). • The **ip vrf** *vrf-name* command creates a VRF routing table and a CEF table, and both are named using the *vrf-name* argument. Associated with these tables is the default route distinguisher value.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:10`	Creates routing and forwarding tables for the VRF instance. • There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).
Step 5	**import ipv4** {**unicast** | **multicast**} [*prefix-limit*] **map** *route-map* **Example:** `Router(config-vrf)# import ipv4 unicast 1000 map UNICAST`	Imports IPv4 prefixes from the global routing table to a VRF table, filtered by the specified route map. • Unicast or multicast prefixes are specified. • Up to a 1000 prefixes will be imported by default. The *prefix-limit* argument is used to specify a limit from 1 to 2,147,483,647 prefixes. • The example references a route map that will import up to 1000 unicast prefixes that pass through the route map.
Step 6	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map UNICAST permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. • The route map name must match the route map specified in Step 5. • The example creates a route map named UNICAST.

	Command or Action	Purpose
Step 8	**match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] | **prefix-list** *prefix-list-name* [*prefix-list-name*]} **Example:** `Router(config-route-map)# match ip address 50`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • Both IP access lists and IP prefix lists are supported. • The example configures the route map to use standard access list 50 to define match criteria.
Step 9	**end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Filtering on the Ingress Interface

The BGP Support for IP Prefix Import from Global Table into a VRF Table feature can be configured globally or on a per-interface basis. We recommend that you apply it to ingress interfaces to maximize performance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip policy route-map** *map-tag*
5. **ip verify unicast vrf** *vrf-name* {**deny** | **permit**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface Ethernet0/0`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip policy route-map** *map-tag* **Example:** `Router(config-if)# ip policy route-map UNICAST`	Identifies a route map to use for policy routing on an interface. • The example attaches the route map named UNICAST to the interface.
Step 5	**ip verify unicast vrf** *vrf-name* {**deny** | **permit**} **Example:** `Router(config-if)# ip verify unicast vrf GREEN permit`	(Optional) Enables Unicast Reverse Path Forwarding verification for the specified VRF. • The example enables verification for the VRF named GREEN. Traffic that passes verification will be forwarded.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Global IP Prefix Import

Perform the steps in this task to display information about the VRFs that are configured with the BGP Support for IP Prefix Import from Global Table into a VRF Table feature and to verify that global IP prefixes are imported into the specified VRF table.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*}
3. **show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device# enable
```

Step 2 **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*}

Displays VPN address information from the BGP table. The output displays the import route map, the traffic type (unicast or multicast), the default or user-defined prefix import limit, the actual number of prefixes that are imported, and individual import prefix entries.

Example:

```
Device# show ip bgp vpnv4 all
```

```
BGP table version is 15, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf academic)
Import Map: ACADEMIC, Address-Family: IPv4 Unicast, Pfx Count/Limit: 6/1000
*> 10.50.1.0/24     172.17.2.2                             0 2 3 ?
*> 10.50.2.0/24     172.17.2.2                             0 2 3 ?
*> 10.50.3.0/24     172.17.2.2                             0 2 3 ?
*> 10.60.1.0/24     172.17.2.2                             0 2 3 ?
*> 10.60.2.0/24     172.17.2.2                             0 2 3 ?
*> 10.60.3.0/24     172.17.2.2                             0 2 3 ?
Route Distinguisher: 200:1 (default for vrf residence)
Import Map: RESIDENCE, Address-Family: IPv4 Unicast, Pfx Count/Limit: 3/1000
*> 10.30.1.0/24     172.17.2.2                  0          0 2 i
*> 10.30.2.0/24     172.17.2.2                  0          0 2 i
*> 10.30.3.0/24     172.17.2.2                  0          0 2 i
Route Distinguisher: 300:1 (default for vrf BLACKHOLE)
Import Map: BLACKHOLE, Address-Family: IPv4 Unicast, Pfx Count/Limit: 3/1000
*> 10.40.1.0/24     172.17.2.2                  0          0 2 i
*> 10.40.2.0/24     172.17.2.2                  0          0 2 i
*> 10.40.3.0/24     172.17.2.2                  0          0 2 i
Route Distinguisher: 400:1 (default for vrf multicast)
Import Map: MCAST, Address-Family: IPv4 Multicast, Pfx Count/Limit: 2/2
*> 10.70.1.0/24     172.17.2.2                  0          0 2 i
*> 10.70.2.0/24     172.17.2.2                  0          0 2 i
```

Step 3 **show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*]

Displays defined VRFs and their associated interfaces. The output displays the import route map, the traffic type (unicast or multicast), and the default or user-defined prefix import limit. The following example output shows that the import route map named UNICAST is importing IPv4 unicast prefixes and that the prefix import limit is 1000.

Example:

```
Device# show ip vrf detail

VRF academic; default RD 100:10; default VPNID <not set>
VRF Table ID = 1
  No interfaces
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:10
  Import VPN route-target communities
    RT:100:10
  Import route-map for ipv4 unicast: UNICAST (prefix limit: 1000)
  No export route-map
```

Configuration Examples for BGP Support for IP Prefix Import from Global Table into a VRF Table

Example: Importing IP Prefixes from Global Table into a VRF Table

The following example imports unicast prefixes into the VRF named *green* by using an IP prefix list and a route map:

This example starts in global configuration mode:

```
!
ip prefix-list COLORADO seq 5 permit 10.131.64.0/19
ip prefix-list COLORADO seq 10 permit 172.31.2.0/30
ip prefix-list COLORADO seq 15 permit 172.31.1.1/32
!
ip vrf green
 rd 200:1
 import ipv4 unicast map UNICAST
 route-target export 200:10
 route-target import 200:10
!
 exit
!
route-map UNICAST permit 10
 match ip address prefix-list COLORADO
!
 exit
```

Example: Verifying IP Prefix Import to a VRF Table

The **show ip vrf** command or the **show ip bgp vpnv4** command can be used to verify that prefixes are imported from the global routing table to the VRF table.

The following sample output shows that the import route map named UNICAST is importing IPv4 unicast prefixes and the prefix import limit is 1000:

```
Device# show ip vrf detail

VRF green; default RD 200:1; default VPNID <not set>
  Interfaces:
    Se2/0
VRF Table ID = 1
  Export VPN route-target communities
    RT:200:10
  Import VPN route-target communities
    RT:200:10
  Import route-map for ipv4 unicast: UNICAST (prefix limit: 1000)
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
VRF red; default RD 200:2; default VPNID <not set>
  Interfaces:
    Se3/0
VRF Table ID = 2
```

```
  Export VPN route-target communities
    RT:200:20
  Import VPN route-target communities
    RT:200:20
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

The following sample output displays the import route map names, the prefix import limit and the actual number of imported prefixes, and the individual import entries:

```
Device# show ip bgp vpnv4 all

BGP table version is 18, local router ID is 10.131.127.252
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:1 (default for vrf green)
Import Map: UNICAST, Address-Family: IPv4 Unicast, Pfx Count/Limit: 1/1000
*>i10.131.64.0/19   10.131.95.252            0    100      0 i
*> 172.16.1.1/32    172.16.2.1               0         32768 i
*> 172.16.2.0/30    0.0.0.0                  0         32768 i
*>i172.31.1.1/32    10.131.95.252            0    100      0 i
*>i172.31.2.0/30    10.131.95.252            0    100      0 i
Route Distinguisher: 200:2 (default for vrf red)
*> 172.16.1.1/32    172.16.2.1               0         32768 i
*> 172.16.2.0/30    0.0.0.0                  0         32768 i
*>i172.31.1.1/32    10.131.95.252            0    100      0 i
*>i172.31.2.0/30    10.131.95.252            0    100      0 i
```

Additional References for Internal BGP Features

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP overview	“Cisco BGP Overview” module
Basic BGP configuration tasks	“Configuring a Basic BGP Network” module
iBGP multipath load sharing	“iBGP Multipath Load Sharing” module
Connecting to a service provider	“Connecting to a Service Provider Using External BGP” module
Configuring features that apply to multiple IP routing protocols	*IP Routing: Protocol-Independent Configuration Guide*

RFCs

RFC	Title
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2519	*A Framework for Inter-Domain Route Aggregation*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 4271	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4893	*BGP Support for Four-octet AS Number Space*
RFC 5396	*Textual Representation of Autonomous system (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

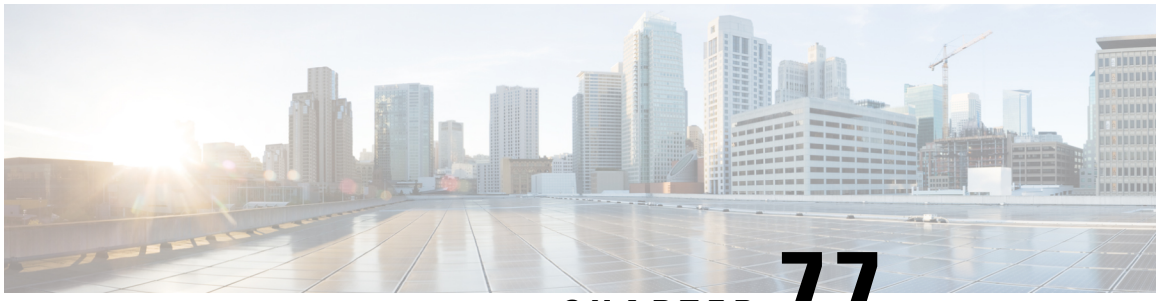
Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 102: Feature Information for BGP Support for IP Prefix Import from Global Table into a VRF Table

Feature Name	Releases	Feature Information
BGP Support for IP Prefix Import from Global Table into a VRF Table	Cisco IOS XE Release 2.1	The BGP Support for IP Prefix Import from Global Table into a VRF Table feature introduces the capability to import IPv4 unicast prefixes from the global routing table into a Virtual Private Network (VPN) routing/forwarding (VRF) instance table using an import route map. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature: **debug ip bgp import**, **import ipv4**, **ip verify unicast vrf**.



CHAPTER 77

BGP Support for IP Prefix Export from a VRF Table into the Global Table

This feature allows a network administrator to export IP prefixes from a VRF table into the global routing table.

Information About IP Prefix Export from a VRF Table into the Global Table

Benefits of IP Prefix Export from a VRF Table into the Global Table

- You can manage some network resources inside a VRF by using a network management node residing in the global table.
- You own some internet public IP address space, but prefer to have a VRF to manage those IP addresses.

How IP Prefix Export from a VRF Table into the Global Table Works

MPLS-VPN using Multiprotocol BGP (MP-BGP) provides a very flexible but secured VPN provisioning mechanism for service providers and customers. However, some customers prefer to relax the boundary so that some specific prefixes can be reachable in a VRF as well as in the global routing table.

Prior to the BGP Support for IP Prefix Export from a VRF Table into Global Table feature, BGP already supported the global-to-VRF import of prefixes. See the "*BGP Support for IP Prefix Import from Global Table into a VRF Table*" module for complete documentation of that feature. Together, the import feature and export feature provide L3VPN dynamic route leaking.

The BGP Support for IP Prefix Export from a VRF Table into the Global Table feature provides the reverse mechanism of the import feature referenced above; it supports the export of prefixes from a VRF table to the

global routing table. It is achieved with an **export {ipv4 | ipv6} {unicast | multicast} map** command, which specifies a route map to control the prefixes that are exported from a VRF table to the global routing table.

Caution The IP Prefix Export from a VRF Table into Global Table feature leaks VRF routes into the global BGP routing table; those routes will be installed into the IPv4 or IPv6 routing table. Use extreme caution to design the network so that such leaking does not affect the normal Internet routing.

Export actions are triggered when a new routing update is received or when routes are withdrawn. During the initial BGP update period, the export action is postponed to allow BGP to converge more quickly. Once BGP converges, incremental BGP updates are evaluated immediately and qualified prefixes are exported as they are received.

Each VRF can export to only one of the global topologies in IPv4 (unicast or multicast) and can export to only one of the global topologies in IPv6 (unicast or multicast).

There is no limit to the number of VRFs per router that can be configured to export IPv4 or IPv6 prefixes to the global routing table.

By default, the software limits the number of prefixes that can be exported per VRF to 1000 prefixes. You can change that limit to a number in the range from 1 to 2,147,483,647 prefixes for each VRF. We recommend that you use caution if you increase the prefix limit above 1000. Configuring the device to export too many prefixes can interrupt normal router operation.

The following **match** and **set** commands are supported in this feature:

- **match as-path**
- **match community** [**exact-match**]
- **match extcommunity**
- **match ip address** [**prefix-list**]
- **match ip next-hop**
- **match ip route-source**
- **match ipv6 address** [**prefix-list**]
- **match ipv6 route-source**
- **match ipv6 next-hop**
- **match policy-list**
- **match route-type**
- **set as-path prepend** [**last-as**]
- **set community additive**
- **set extcommunity** [**cost** | **rt**]
- **set extcomm-list delete**
- **set ip next-hop**
- **set ipv6 next-hop**

- **set local-preference**
- **set metric**
- **set origin**
- **set weight**

Note The **set ip vrf next-hop** and **set ipv6 vrf next-hop** commands are not supported in this feature.

How to Export IP Prefixes from a VRF Table into the Global Table

Creating the VRF and the Export Route Map for an Address Family

The IP prefixes that are defined for export are processed through a match clause in a route map. IP prefixes that pass through the route map are exported into the global routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **export** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*prefix-limit*] **map** *map-name*
7. **route-target import** *route-target-ext-community*
8. **route-target export** *route-target-ext-community*
9. **exit**
10. **exit**
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] | **prefix-list** *prefix-list-name* [*prefix-list-name*]}
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vpn1`	Creates a VRF routing table and specifies the VRF name (or tag).
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:100`	Creates routing and forwarding tables for the VRF instance. • There are two formats for configuring the argument. It can be configured in the *as-number*:*network number* (*ASN*:*nn*) format, as shown in the example, or it can be configured in the *IP address*:*network number* format (*IP-address*:*nn*).
Step 5	**address-family** {**ipv4** | **ipv6**} **Example:** `Device(config-vrf)# address-family ipv4`	Configures the IPv4 or IPv6 address family.
Step 6	**export** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*prefix-limit*] **map** *map-name* **Example:** `Device(config-vrf-af)# export ipv4 unicast 500 map UNICAST`	Exports IPv4 or IPv6 prefixes from the VRF table to the global routing table, filtered by the specified route map. • Specify **ipv4** or **ipv6**, which you specified in Step 5. This example exports IPv4 unicast prefixes. • Based on this example, no more than 500 prefixes will be exported. • The prefixes exported are those that pass the route map.
Step 7	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 100:100`	Creates a route-target extended community for a VRF instance. • For information about route-target import or export, see the *MPLS: Layer 3 VPNs Configuration Guide*.
Step 8	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 100:100`	Creates a route-target extended community for a VRF instance.
Step 9	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits address family configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 11	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map UNICAST permit 10`	Enables policy routing. • The example creates a route map named UNICAST.
Step 12	**match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] | **prefix-list** *prefix-list-name* [*prefix-list-name*]} **Example:** `Device(config-route-map)# match ip address 50`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • Both IP access lists and IP prefix lists are supported. • The example configures the route map to use standard access list 50 to define match criteria. • Define the access list (not shown in this task); for example, access-list 50 permit 192.168.1.0 255.255.255.0.
Step 13	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Creating the VRF and the Export Route Map for a VRF (IPv4 only)

The IP prefixes that are defined for export are processed through a match clause in a route map. IP prefixes that pass through the route map are exported into the global routing table.

Note
- Only IPv4 unicast and multicast prefixes can be exported from a VRF table to the global routing table under the **ip vrf** command, as shown in this task. To export IPv6 prefixes, you must do so under the IPv6 address family; see the section "Creating the VRF and the Export Route Map Per Address Family."
- IPv4 prefixes exported into the global routing table using this feature cannot be exported into a VPNv4 VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*

5. **export ipv4** {**unicast** | **multicast**} [*prefix-limit*] **map** *map-tag*
6. **route-target import** *route-target-ext-community*
7. **route-target export** *route-target-ext-community*
8. **exit**
9. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
10. **match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] | **prefix-list** *prefix-list-name* [*prefix-list-name*]}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf GREEN`	Creates a VRF routing table and specifies the VRF name (or tag). • The **ip vrf** *vrf-name* command creates a VRF routing table and a CEF table, and both are named using the *vrf-name* argument. Associated with these tables is the default route distinguisher value.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:10`	Creates routing and forwarding tables for the VRF instance. • There are two formats for configuring the argument. It can be configured in the *as-number***:***network number* (*ASN***:***nn*) format, as shown in the example, or it can be configured in the *IP-address***:***network number* format (*IP-address***:***nn*).
Step 5	**export ipv4** {**unicast**	**multicast**} [*prefix-limit*] **map** *map-tag* **Example:** `Device(config-vrf)# export ipv4 unicast 500 map UNICAST`	Exports IPv4 prefixes from the VRF table to the global routing table, filtered by the specified route map. • Unicast or multicast prefixes are specified. • By default, up to 1000 prefixes can be exported. The *prefix-limit* argument is used to specify a limit from 1 to 2,147,483,647 prefixes. • The example creates an export map that will export up to 500 unicast prefixes that pass through the route map named UNICAST.

	Command or Action	Purpose
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:100`	Creates a route-target extended community for a VRF instance. • For information about route-target import or export, see the *MPLS: Layer 3 VPNs Configuration Guide*.
Step 7	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 100:100`	Creates a route-target extended community for a VRF instance.
Step 8	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map UNICAST permit 10`	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. • The route map name must match the route map specified in Step 5. • The example creates a route map named UNICAST.
Step 10	**match ip address** {*acl-number* [*acl-number* | *acl-name*] | *acl-name* [*acl-name* | *acl-number*] | **prefix-list** *prefix-list-name* [*prefix-list-name*]} **Example:** `Device(config-route-map)# match ip address 50`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, and performs policy routing on matched packets. • Both IP access lists and IP prefix lists are supported. • The example configures the route map to use standard access list 50 to define match criteria.
Step 11	**end** **Example:** `Device(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Displaying Information About IP Prefix Export from a VRF into the Global Table

Perform any of the steps in this task to see information about the prefixes exported from a VRF table into the global table.

SUMMARY STEPS

1. **enable**
2. **show ip bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*prefix*]
3. **debug ip bgp import event**

4. **debug ip bgp import update**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*prefix*] **Example:** `Device# show ip bgp ipv4 unicast 192.168.1.1`	Displays information about the imported path from a VRF to the global table.
Step 3	**debug ip bgp import event** **Example:** `Device# debug ip bgp import event`	Displays messages related to IPv4 prefix import events.
Step 4	**debug ip bgp import update** **Example:** `Device# debug ip bgp import update`	Displays messages related to IPv4 prefix import updates.

Configuration Examples for IP Prefix Export from a VRF Table into the Global Table

Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IPv6 Address Family

```
vrf definition X
  rd 100:100
   address-family ipv6
    export ipv6 unicast map OnlyNet2000
    route-target import 100:100
    route-target export 100:100
!
 ipv6 prefix-list net2000 permit 2000::/16
!
route-map OnlyNet2000 permit 10
 match ipv6 address prefix-list net2000
```

Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IPv4 Address Family

```
vrf definition X
  rd 100:100
   address-family ipv4
    export ipv4 unicast map OnlyNet200
    route-target import 100:100
    route-target export 100:100
!
 ip prefix-list net200 permit 200.0.0.0/8
!
route-map OnlyNet200 permit 10
 match ip address prefix-list net200
```

Example: Exporting IP Prefixes from a VRF Table into the Global Table Using IP VRF (IPv4 Only)

```
 ip vrf vrfname
  rd 100:100
  export ipv4 unicast map OnlyNet200
  route-target import 100:100
  route-target export 100:100
!
 ip prefix-list net200 permit 200.0.0.0/8
!
route-map OnlyNet200 permit 10
 match ip address prefix-list net200
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS BGP Command Reference
Use of route-target import and export	*MPLS: Layer 3 VPNs Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

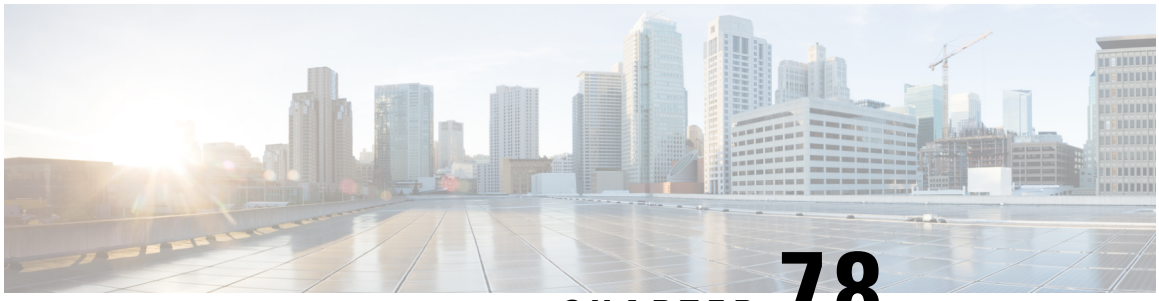
Feature Information for IP Prefix Export from a VRF Table into the Global Table

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 103: Feature Information for BGP Support for IP Prefix Export from a VRF Table into the Global Table

Feature Name	Releases	Feature Information
BGP Support for IP Prefix Export from a VRF Table into the Global Table		This feature allows a network administrator to export IP prefixes from a VRF routing table into the global routing table. The following command was introduced: **export map (VRF table to global table)**. The following commands were modified: **debug ip bgp import** and **show ip bgp**.



CHAPTER 78

BGP per Neighbor SoO Configuration

The BGP per Neighbor SoO Configuration feature simplifies the configuration of the site-of-origin (SoO) value. Per neighbor SoO configuration introduces two new commands that can be configured in submodes under router configuration mode to set the SoO value.

Prerequisites for BGP per Neighbor SoO Configuration

This feature assumes that a Border Gateway Protocol (BGP) network is configured and that Cisco Express Forwarding is enabled in your network.

Restrictions for BGP per Neighbor SoO Configuration

A BGP neighbor or peer policy template-based SoO configuration takes precedence over the SoO value configured in an inbound route map.

Information About Configuring BGP per Neighbor SoO

Site of Origin BGP Community Attribute

The site-of-origin (SoO) extended community is a BGP extended community attribute that is used to identify routes that have originated from a site so that the readvertisement of that prefix back to the source site can be prevented. The SoO extended community uniquely identifies the site from which a router has learned a route. BGP can use the SoO value associated with a route to prevent routing loops.

Route Distinguisher

A route distinguisher (RD) creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of an IPv4 prefix to change it into a globally unique VPN-IPv4 prefix. An RD can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter an RD in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:

45000:3

- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:

192.168.10.15:1

BGP per Neighbor Site of Origin Configuration

There are three ways to configure an SoO value for a BGP neighbor:

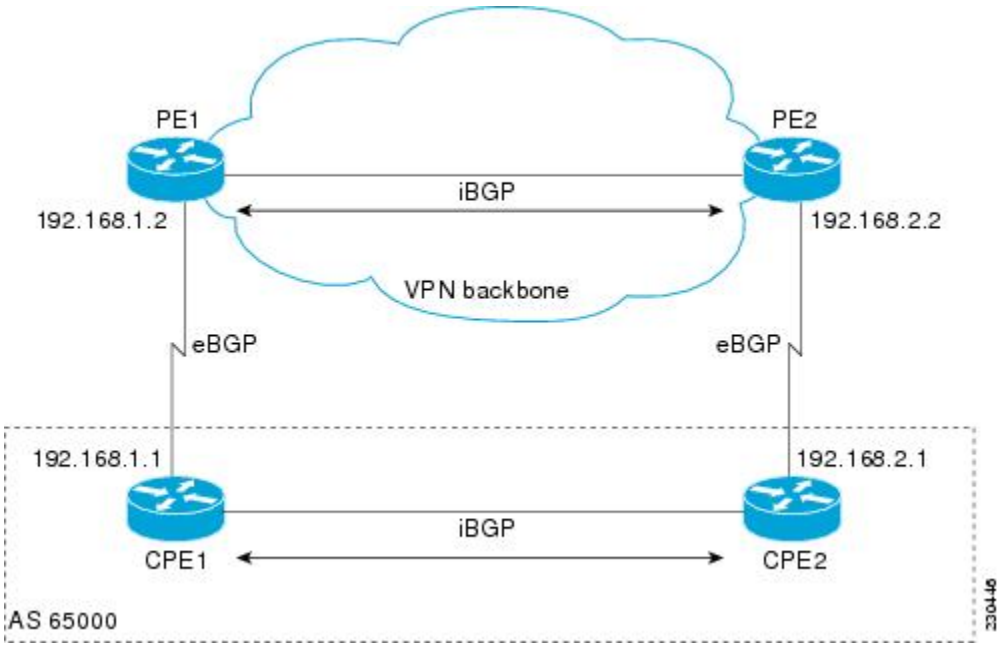
- BGP peer policy template--A peer policy template is created, and an SoO value is configured as part of the peer policy. Under address family IPv4 VRF, a neighbor is identified and is configured to inherit the peer policy that contains the SoO value.
- BGP **neighbor** command--Under address family IPv4 VRF, a neighbor is identified, and an SoO value is configured for the neighbor.
- BGP peer group--Under address family IPv4 VRF, a BGP peer group is configured, an SoO value is configured for the peer group, a neighbor is identified, and the neighbor is configured as a member of the peer group.

Note A BGP neighbor or peer policy template-based SoO configuration takes precedence over the SoO value configured in an inbound route map.

The configuration of SoO values for BGP neighbors is performed on a provider edge (PE) router, which is the VPN entry point. When SoO is enabled, the PE router forwards prefixes to the customer premises equipment (CPE) only when the SoO tag of the prefix does not match the SoO tag configured for the CPE.

For example, in the figure below, an SoO tag is set as 65000:1 for the customer site that includes routers CPE1 and CPE2 with an autonomous system number of 65000. When CPE1 sends prefixes to PE1, PE1 tags the prefixes with 65000:1, which is the SoO tag for CPE1 and CPE2. When PE1 sends the tagged prefixes to PE2, PE2 performs a match against the SoO tag from CPE2. Any prefixes with the tag value of 65000:1 are not sent to CPE2 because the SoO tag matches the SoO tag of CPE2, and a routing loop is avoided.

Figure 93: Network Diagram for SoO Example



Benefits of BGP per Neighbor Site of Origin

In releases prior to the introduction of this feature, the SoO extended community attribute is configured using an inbound route map that sets the SoO value during the update process. With the introduction of the BGP per Neighbor Site of Origin feature, two new commands configured in submodes under router configuration mode simplify the SoO value configuration.

BGP Peer Policy Templates

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address families. Peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through inheritance from peer policy templates. The configuration of peer policy templates simplifies the configuration of BGP policy commands that are applied to all neighbors within an autonomous system.

Peer policy templates support inheritance. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. So, a total of eight peer policy templates can be applied to a neighbor or neighbor group.

The configuration of peer policy templates simplifies and improves the flexibility of BGP configuration. A specific policy can be configured once and referenced many times. Because a peer policy supports up to eight levels of inheritance, very specific and very complex BGP policies can be created.

For more details about BGP peer policy templates, see the "Configuring a Basic BGP Network" module.

How to Configure BGP per Neighbor SoO

Enabling Cisco Express Forwarding and Configuring VRF Instances

Perform this task on both of the PE routers in the figure above to configure Virtual Routing and Forwarding (VRF) instances to be used with the per-VRF assignment tasks. In this task, Cisco Express Forwarding is enabled, and a VRF instance named SOO_VRF is created. To make the VRF functional, a route distinguisher is created, and the VRF is associated with an interface. When the route distinguisher is created, the routing and forwarding tables are created for the VRF instance named SOO_VRF. After associating the VRF with an interface, the interface is configured with an IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef**
4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**export** | **both**} *route-target-ext-community*
7. **route-target** {**import** | **both**} *route-target-ext-community*
8. **exit**
9. **interface** *type number*
10. **ip vrf forwarding** *vrf-name* [**downstream** *vrf-name2*]
11. **ip address** *ip-address mask* [**secondary**]
12. **end**
13. **show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*] [*output-modifiers*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** **Example:** `Device(config)# ip cef`	Enables Cisco Express Forwarding on the route processor.

	Command or Action	Purpose
Step 4	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf SOO_VRF`	Defines a VRF instance and enters VRF configuration mode.

Step 5	**rd** *route-distinguisher*
	Example:
	`Device(config-vrf)# rd 1:1`
		Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN.
		• Use the *route-distinguisher* argument to specify the default RD for a VPN. There are two formats that you can use to specify an RD:
		  • A 16-bit autonomous system number, a colon, and a 32-bit number, for example: 65000:3
		  • A 32-bit IP address, a colon, and a 16-bit number, for example: 192.168.1.2:51
		• In this example, the RD uses an autonomous system number with the number 1 after the colon.

Step 6	**route-target** {**export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 1:1`	Creates a route-target extended community for a VRF. • Use the **export** keyword to export routing information to the target VPN extended community. • Use the **both** keyword to both import routing information from, and export routing information to, the target VPN extended community. • Use the *route-target-ext-community* argument to specify the VPN extended community. **Note** Only the syntax applicable to this step is displayed. For a different use of this syntax, see Step 7.
Step 7	**route-target** {**import** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 1:1`	Creates a route-target extended community for a VRF. • Use the **import** keyword to import routing information from the target VPN extended community. • Use the **both** keyword to both import routing information from, and export routing information to, the target VPN extended community. • Use the *route-target-ext-community* argument to specify the VPN extended community.
Step 8	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 9	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 10	**ip vrf forwarding** *vrf-name* [**downstream** *vrf-name2*] **Example:** `Device(config-if)# ip vrf forwarding SOO_VRF`	Associates a VRF with an interface or subinterface. • In this example, the VRF named SOO_VRF is associated with Gigabit Ethernet interface 1/0/0. **Note** Executing this command on an interface removes the IP address, so the IP address should be reconfigured.
Step 11	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 192.168.1.2 255.255.255.0`	Configures an IP address. • In this example, Gigabit Ethernet interface 1/0/0 is configured with an IP address of 192.168.1.2.
Step 12	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 13	**show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*] [*output-modifiers*] **Example:** `Device# show ip vrf`	Displays the configured VRFs. • Use this command to verify the configuration of this task.

Examples

The following output of the **show ip vrf** command displays the VRF named SOO_VRF configured in this task.

```
Device# show ip vrf

Name                              Default RD          Interfaces
SOO_VRF                                1:1              GE1/0/0
```

Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template

Perform this task on router PE1 in the figure above to configure an SoO value for a BGP neighbor at the router CPE1 in the figure above using a peer policy template. In this task, a peer policy template is created, and the SoO value is configured for the peer policy. Under address family IPv4 VRF, a neighbor is identified and is configured to inherit the peer policy that contains the SoO value.

If a BGP peer inherits from several peer policy templates that specify different SoO values, the SoO value in the last template applied takes precedence and is applied to the peer. However, direct configuration of the SoO value on the BGP neighbor overrides any inherited template configurations of the SoO value.

Before you begin

This task assumes that the task described in the Enabling Cisco Express Forwarding and Configuring VRF Instances, on page 1164 has been performed.

Note A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **soo** *extended-community-value*
6. **exit-peer-policy**
7. **address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*]
8. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
9. **neighbor** *ip-address* **activate**
10. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 50000`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 4	**template peer-policy** *policy-template-name* **Example:** `Router(config-router)# template peer-policy SOO_POLICY`	Creates a peer policy template and enters policy-template configuration mode.

Step 5	**soo** *extended-community-value*
	Example:
	`Router(config-router-ptmp)# soo 65000:1`
		Sets the SoO value for a BGP peer policy template.
		• Use the *extended-community-value* argument to specify the VPN extended community value. The value takes one of the following formats:
		  • A 16-bit autonomous system number, a colon, and a 32-bit number, for example: 45000:3
		  • A 32-bit IP address, a colon, and a 16-bit number, for example: 192.168.10.2:51
		• In this example, the SoO value is set at 65000:1.

Step 6	**exit-peer-policy** **Example:** `Router(config-router-pmtp)# exit-peer-policy`	Exits policy-template configuration mode and returns to router configuration mode.
Step 7	**address-family ipv4** [**unicast** | **multicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf SOO_VRF`	Specifies the IPv4 address family and enters address family configuration mode. • Use the **unicast** keyword to specify the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 8	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 65000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 9	**neighbor** *ip-address* **activate** **Example:**	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor 192.168.1.1 activate`	
Step 10	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 inherit peer-policy SOO_POLICY`	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration. • In this example, the router is configured to send the peer policy template named SOO_POLICY to the 192.168.1.1 neighbor to inherit. If another peer policy template is indirectly inherited from SOO_POLICY, the indirectly inherited configuration will also be applied. Up to seven additional peer policy templates can be indirectly inherited from SOO_POLICY.
Step 11	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring a per Neighbor SoO Value Using a BGP neighbor Command

Perform this task on router PE2 in the figure above to configure an SoO value for the BGP neighbor at router CPE2 in the figure above using a **neighbor** command. For the IPv4 VRF address family, a neighbor is identified, and an SoO value is configured for the neighbor.

Direct configuration of the SoO value on a BGP neighbor overrides any inherited peer policy template configurations of the SoO value.

Before you begin

This task assumes that the task described in the "Verifying CEF and Configuring VRF Instances" section has been performed with appropriate changes to interfaces and IP addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **activate**
7. **neighbor** {*ip-address* | *peer-group-name*} **soo** *extended-community-value*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf SOO_VRF`	Specifies the IPv4 address family and enters address family configuration mode. • Use the **unicast** keyword to specify the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.2.1 remote-as 65000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 6	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.2.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router. • In this example, the external BGP peer at 192.168.2.1 is activated.

	Command or Action	Purpose
		Note If a peer group has been configured in Step 5 , do not use this step because BGP peer groups are activated when any parameter is configured. For example, a BGP peer group is activated when an SoO value is configured using the **neighbor soo** command in Step 7.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **soo** *extended-community-value* **Example:** `Device(config-router-af)# neighbor 192.168.2.1 soo 65000:1`	Sets the site-of-origin (SoO) value for a BGP neighbor or peer group. • In this example, the neighbor at 192.168.2.1 is configured with an SoO value of 65000:1.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring a per Neighbor SoO Value Using a BGP Peer Group

Perform this task on router PE1 in the figure above to configure an SoO value for the BGP neighbor at router CPE1 in the figure above using a **neighbor** command with a BGP peer group. Under address family IPv4 VRF, a BGP peer group is created and an SoO value is configured using a BGP **neighbor** command, and a neighbor is then identified and added as a peer group member. A BGP peer group member inherits the configuration associated with a peer group, which in this example, includes the SoO value.

Direct configuration of the SoO value on a BGP neighbor overrides any inherited peer group configurations of the SoO value.

Before you begin

This task assumes that the task described in "Enabling Cisco Express Forwarding and Configuring VRF Instances" has been performed.

Note A BGP peer cannot inherit from a peer policy or session template and be configured as a peer group member at the same. BGP templates and BGP peer groups are mutually exclusive.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** {*ip-address* | *peer-group-name*} **soo** *extended-community-value*

7. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
8. **neighbor** *ip-address* **activate**
9. **neighbor** *ip-address* **peer-group** *peer-group-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf SOO_VRF`	Specifies the IPv4 address family and enters address family configuration mode. • Use the **unicast** keyword to specify the IPv4 unicast address family. By default, the router is placed in configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router-af)# neighbor SOO_group peer-group`	Creates a BGP peer group.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **soo** *extended-community-value* **Example:**	Sets the site-of-origin (SoO) value for a BGP neighbor or peer group. • In this example, the BGP peer group, SOO_group, is configured with an SoO value of 65000:1.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor SOO_group soo 65000:1`	
Step 7	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 65000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 8	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 VRF address family with the local router.
Step 9	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 peer-group SOO_group`	Assigns the IP address of a BGP neighbor to a peer group.
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP per Neighbor SoO Configuration

Example: Configuring a per Neighbor SoO Value Using a BGP Peer Policy Template

The following example shows how to create a peer policy template and configure an SoO value as part of the peer policy. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a peer policy template is created and an SoO value is configured as part of the peer policy. Under the IPv4 VRF address family, a neighbor is identified and configured to inherit the peer policy that contains the SoO value.

```
ip cef
ip vrf SOO_VRF
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 exit
interface GigabitEthernet 1/0/0
 ip vrf forwarding SOO_VRF
 ip address 192.168.1.2 255.255.255.0
 exit
```

```
router bgp 50000
 template peer-policy SOO_POLICY
  soo 65000:1
  exit-peer-policy
 address-family ipv4 vrf SOO_VRF
  neighbor 192.168.1.1 remote-as 65000
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 inherit peer-policy SOO_POLICY
  end
```

Example: Configuring a per Neighbor SoO Value Using a BGP neighbor Command

The following example shows how to configure an SoO value for a BGP neighbor. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a neighbor is identified in the IPv4 VRF address family and an SoO value is configured for the neighbor.

```
ip cef
ip vrf SOO_VRF
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 exit
interface GigabitEthernet 1/0/0
 ip vrf forwarding SOO_VRF
 ip address 192.168.2.2 255.255.255.0
 exit
router bgp 50000
 address-family ipv4 vrf SOO_VRF
  neighbor 192.168.2.1 remote-as 65000
  neighbor 192.168.2.1 activate
  neighbor 192.168.2.1 soo 65000:1
  end
```

Example: Configuring a per Neighbor SoO Value Using a BGP Peer Group

The following example shows how to configure an SoO value for a BGP peer group. After enabling Cisco Express Forwarding and configuring a VRF instance named SOO_VRF, a BGP peer group is configured in the IPv4 VRF address family, an SoO value is configured for the peer group, a neighbor is identified, and the neighbor is configured as a member of the peer group.

```
ip cef
ip vrf SOO_VRF
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 exit
interface GigabitEthernet 1/0/0
 ip vrf forwarding SOO_VRF
 ip address 192.168.1.2 255.255.255.0
 exit
router bgp 50000
 address-family ipv4 vrf SOO_VRF
  neighbor SOO_GROUP peer-group
  neighbor SOO_GROUP soo 65000:65
  neighbor 192.168.1.1 remote-as 65000
  neighbor 192.168.1.1 activate
```

```
neighbor 192.168.1.1 peer-group SOO_GROUP
end
```

Where to Go Next

- To read an overview of BGP, proceed to the "Cisco BGP Overview" module.
- To perform basic BGP feature tasks, proceed to the "Configuring a Basic BGP Network" module.
- To perform advanced BGP feature tasks, proceed to the "Configuring Advanced BGP Features" module.
- To configure BGP neighbor session options, proceed to the "Configuring BGP Neighbor Session Options" module.
- To perform internal BGP tasks, proceed to the "Configuring Internal BGP Features" module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
IP Switching commands	Cisco IOS IP Switching Command Reference

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

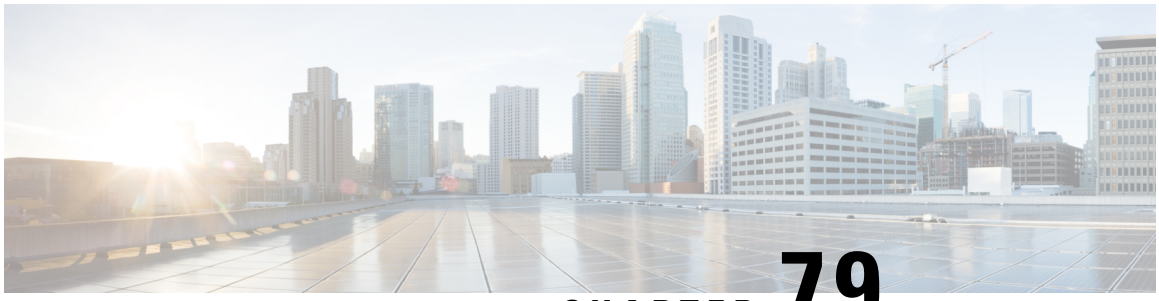
Feature Information for BGP per Neighbor SoO Configuration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 104: Feature Information for BGP per Neighbor SoO Configuration

Feature Name	Releases	Feature Information
BGP per Neighbor SoO Configuration	Cisco IOS XE Release 2.1	The BGP per neighbor SOO configuration feature simplifies the configuration of the site-of-origin (SoO) parameter. The per neighbor SoO configuration introduces two new commands that can be configured in submodes under router configuration mode to set the SoO value. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced by this feature: **neighbor soo**, **soo**.



CHAPTER 79

Per-VRF Assignment of BGP Router ID

The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP) on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing **bgp router-id** command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF.

Prerequisites for Per-VRF Assignment of BGP Router ID

Before you configure this feature, Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled in the network, and basic BGP peering is assumed to be running in the network.

Information About Per-VRF Assignment of BGP Router ID

BGP Router ID

The BGP router identifier (ID) is a 4-byte field that is set to the highest IP address on the router. Loopback interface addresses are considered before physical interface addresses because loopback interfaces are more stable than physical interfaces. The BGP router ID is used in the BGP algorithm for determining the best path to a destination where the preference is for the BGP router with the lowest router ID. It is possible to manually configure the BGP router ID using the **bgp router-id** command to influence the best path algorithm.

Per-VRF Router ID Assignment

In Cisco IOS XE Release 2.1 and later releases, support for configuring separate router IDs for each Virtual Private Network (VPN) routing/forwarding (VRF) instance was introduced. The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP)

on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing **bgp router-id** command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF.

Route Distinguisher

A route distinguisher (RD) creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of an IPv4 prefix to change it into a globally unique VPN-IPv4 prefix. An RD can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter an RD in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:

45000:3

- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:

192.168.10.15:1

How to Configure Per-VRF Assignment of BGP Router ID

Configuring VRF Instances

Perform this task to configure VRF instances to be used with the per-VRF assignment tasks. In this task, a VRF instance named vrf_trans is created. To make the VRF functional, a route distinguisher is created. When the route distinguisher is created, the routing and forwarding tables are created for the VRF instance named vrf_trans.

Before you begin

This task assumes that you have Cisco Express Forwarding or distributed Cisco Express Forwarding enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **both**} *route-target-ext-community*
6. **route-target** {**export** | **both**} *route-target-ext-community*
7. **exit**
8. Repeat Step 3 through Step 7 for each VRF to be defined.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf_trans`	Defines a VRF instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 45000:2`	Creates routing and forwarding tables for a VRF and specifies the default RD for a VPN. • Use the *route-distinguisher* argument to specify the default RD for a VPN. There are two formats you can use to specify an RD. For more details, see the "Route Distinguisher" section. • In this example, the RD uses an autonomous system number with the number 2 after the colon.
Step 5	**route-target** {**import** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 55000:5`	Creates a route-target extended community for a VRF. • Use the **import** keyword to import routing information from the target VPN extended community. • Use the **both** keyword to both import routing information from and export routing information to the target VPN extended community. • Use the *route-target-ext-community* argument to specify the VPN extended community.
Step 6	**route-target** {**export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 55000:1`	Creates a route-target extended community for a VRF. • Use the **export** keyword to export routing information to the target VPN extended community. • Use the **both** keyword to both import routing information from and export routing information to the target VPN extended community. • Use the *route-target-ext-community* argument to specify the VPN extended community.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 8	Repeat Step 3 through Step 7 for each VRF to be defined.	--

Associating VRF Instances with Interfaces

Perform this task to associate VRF instances with interfaces to be used with the per-VRF assignment tasks. In this task, a VRF instance named vrf_trans is associated with a serial interface.

Make a note of the IP addresses for any interface to which you want to associate a VRF instance because the **ip vrf forwarding** command removes the IP address. Step 8 allows you to reconfigure the IP address.

Before you begin

- This task assumes that you have Cisco Express Forwarding or distributed Cisco Express Forwarding enabled.
- This task assumes that VRF instances have been configured in the Configuring VRF Instances, on page 1178.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **exit**
6. **interface** *type number*
7. **ip vrf forwarding** *vrf-name* [**downstream** *vrf-name2*]
8. **ip address** *ip-address mask* [**secondary**]
9. Repeat Step 5 through Step 8 for each VRF to be associated with an interface.
10. **end**
11. **show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface loopback0`	Configures an interface type and enters interface configuration mode. • In this example, loopback interface 0 is configured.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 172.16.1.1 255.255.255.255`	Configures an IP address. • In this example, the loopback interface is configured with an IP address of 172.16.1.1.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface serial2/0/0`	Configures an interface type and enters interface configuration mode. • In this example, serial interface 2/0/0 is configured.
Step 7	**ip vrf forwarding** *vrf-name* [**downstream** *vrf-name2*] **Example:** `Router(config-if)# ip vrf forwarding vrf_trans`	Associates a VRF with an interface or subinterface. • In this example, the VRF named vrf_trans is associated with serial interface 2/0/0. **Note** Executing this command on an interface removes the IP address. The IP address should be reconfigured.
Step 8	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.4.1 255.255.255.0`	Configures an IP address. • In this example, serial interface 2/0/0 is configured with an IP address of 192.168.4.1.
Step 9	Repeat Step 5 through Step 8 for each VRF to be associated with an interface.	--
Step 10	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 11	**show ip vrf** [**brief** | **detail** | **interfaces** | **id**] [*vrf-name*] **Example:** `Router# show ip vrf interfaces`	(Optional) Displays the set of defined VRFs and associated interfaces. • In this example, the output from this command shows the VRFs that have been created and their associated interfaces.

Examples

The following output shows that two VRF instances named vrf_trans and vrf_users were configured on two serial interfaces.

```
Router# show ip vrf interfaces
Interface          IP-Address      VRF                    Protocol
Serial2            192.168.4.1     vrf_trans              up
Serial3            192.168.5.1     vrf_user               up
```

Manually Configuring a BGP Router ID per VRF

Perform this task to manually configure a BGP router ID for each VRF. In this task, several address family configurations are shown and the router ID is configured in the IPv4 address family mode for one VRF instance. Step 22 shows you how to repeat certain steps to permit the configuration of more than one VRF on the same router.

Before you begin

This task assumes that you have previously created the VRF instances and associated them with interfaces. For more details, see the Configuring VRF Instances, on page 1178 and the Associating VRF Instances with Interfaces, on page 1180.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **bgp log-neighbor-changes**
6. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
7. **neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number*
8. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
9. **neighbor** {*ip-address*| *peer-group-name*} **activate**
10. **neighbor** {*ip-address*| *peer-group-name*} **send-community**{**both**| **standard**| **extended**}
11. **exit-address-family**
12. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
13. **redistribute connected**
14. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*

15. **neighbor** *ip-address* **local-as** *autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]
16. **neighbor** {*ip-address*| *peer-group-name*} **ebgp-multihop**[*ttl*]
17. **neighbor** {*ip-address*| *peer-group-name*} **activate**
18. **neighbor** *ip-address* **allowas-in** [*number*]
19. **no auto-summary**
20. **no synchronization**
21. **bgp router-id** {*ip-address*| **auto-assign**}
22. Repeat Step 11 to Step 21 to configure another VRF instance.
23. **end**
24. **show ip bgp vpnv4** {**all**| **rd** *route-distinguisher*| **vrf** *vrf-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast**router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.
Step 5	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.

	Command or Action	Purpose
Step 6	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor is an internal neighbor.
Step 7	**neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 update-source loopback0`	Allows BGP sessions to use any operational interface for TCP connections. • In this example, BGP TCP connections for the specified neighbor are sourced with the IP address of the loopback interface rather than the best local address.
Step 8	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. • The example creates a VPNv4 address family session.
Step 9	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.1.1 activate`	Activates the neighbor under the VPNv4 address family. • In this example, the neighbor 172.16.1.1 is activated.
Step 10	**neighbor** {*ip-address*| *peer-group-name*} **send-community**{**both**| **standard**| **extended**} **Example:** `Router(config-router-af)# neighbor 172.16.1.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 172.16.1.1.
Step 11	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 12	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Router(config-router)# address-family ipv4 vrf vrf_trans`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. • The example specifies that the VRF instance named vrf_trans is to be associated with subsequent IPv4 address family configuration commands.
Step 13	**redistribute connected** **Example:** `Router(config-router-af)# redistribute connected`	Redistributes from one routing domain into another routing domain. • In this example, the **connected** keyword is used to represent routes that are established automatically when IP is enabled on an interface. • Only the syntax applicable to this step is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 14	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 192.168.1.1 is an external neighbor.
Step 15	**neighbor** *ip-address* **local-as** *autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 local-as 50000 no-prepend`	Customizes the AS_PATH attribute for routes received from an eBGP neighbor. • The autonomous system number from the local BGP routing process is prepended to all external routes by default. • Use the **no-prepend** keyword to not prepend the local autonomous system number to any routes received from the eBGP neighbor. • In this example, routes from the neighbor at 192.168.1.1 will not contain the local autonomous system number.

	Command or Action	Purpose
Step 16	**neighbor** {*ip-address*| *peer-group-name*} **ebgp-multihop**[*ttl*] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 ebgp-multihop 2`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. • In this example, BGP is configured to allow connections to or from neighbor 192.168.1.1, which resides on a network that is not directly connected.
Step 17	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Activates the neighbor under the IPV4 address family. • In this example, the neighbor 192.168.1.1 is activated.
Step 18	**neighbor** *ip-address* **allowas-in** [*number*] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 allowas-in 1`	Configures provider edge (PE) routers to allow the readvertisement of all prefixes that contain duplicate autonomous system numbers. • In the example, the PE router with autonomous system number 45000 is configured to allow prefixes from the VRF vrf-trans. The neighboring PE router with the IP address 192.168.1.1 is set to be readvertised once to other PE routers with the same autonomous system number.
Step 19	**no auto-summary** **Example:** `Router(config-router-af)# no auto-summary`	Disables automatic summarization and sends subprefix routing information across classful network boundaries.
Step 20	**no synchronization** **Example:** `Router(config-router-af)# no synchronization`	Enables the Cisco IOS XE software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP).
Step 21	**bgp router-id** {*ip-address*| **auto-assign**} **Example:** `Router(config-router-af)# bgp router-id 10.99.1.1`	Configures a fixed router ID for the local BGP routing process. • In this example, the specified BGP router ID is assigned for the VRF instance associated with this IPv4 address family configuration.
Step 22	Repeat Step 11 to Step 21 to configure another VRF instance.	--
Step 23	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 24	**show ip bgp vpnv4** {**all**| **rd** *route-distinguisher*| **vrf** *vrf-name*} **Example:** `Router# show ip bgp vpnv4 all`	(Optional) Displays VPN address information from the BGP table. • In this example, the complete VPNv4 database is displayed. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS Multiprotocol Label Switching Command Reference*.

Examples

The following sample output assumes that two VRF instances named vrf_trans and vrf_user were configured each with a separate router ID. The router ID is shown next to the VRF name.

```
Router# show ip bgp vpnv4 all
BGP table version is 5, local router ID is 172.17.1.99
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf vrf_trans) VRF Router ID 10.99.1.2
*> 192.168.4.0      0.0.0.0                  0         32768 ?
Route Distinguisher: 42:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
*> 192.168.5.0      0.0.0.0                  0         32768 ?
```

Automatically Assigning a BGP Router ID per VRF

Perform this task to automatically assign a BGP router ID for each VRF. In this task, a loopback interface is associated with a VRF and the **bgp router-id** command is configured at the router configuration level to automatically assign a BGP router ID to all VRF instances. Step 9 shows you how to repeat certain steps to configure each VRF that is to be associated with an interface. Step 30 shows you how to configure more than one VRF on the same router.

Before you begin

This task assumes that you have previously created the VRF instances. For more details, see the Configuring VRF Instances, on page 1178.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **exit**
6. **interface** *type number*

7. **ip vrf forwarding** *vrf-name* [**downstream** *vrf-name2*]
8. **ip address** *ip-address mask* [**secondary**]
9. Repeat Step 5 through Step 8 for each VRF to be associated with an interface.
10. **exit**
11. **router bgp** *autonomous-system-number*
12. **bgp router-id** {*ip-address*| **vrf auto-assign**}
13. **no bgp default ipv4-unicast**
14. **bgp log-neighbor-changes**
15. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
16. **neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number*
17. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
18. **neighbor** {*ip-address*| *peer-group-name*} **activate**
19. **neighbor** {*ip-address*| *peer-group-name*} **send-community**{**both**| **standard**| **extended**}
20. **exit-address-family**
21. **address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]}
22. **redistribute connected**
23. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
24. **neighbor** *ip-address* **local-as** *autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]]
25. **neighbor** {*ip-address*| *peer-group-name*} **ebgp-multihop**[*ttl*]
26. **neighbor** {*ip-address*| *peer-group-name*} **activate**
27. **neighbor** *ip-address* **allowas-in** [*number*]
28. **no auto-summary**
29. **no synchronization**
30. Repeat Step 20 to Step 29 to configure another VRF instance.
31. **end**
32. **show ip bgp vpnv4** {**all**| **rd** *route-distinguisher*| **vrf** *vrf-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface loopback0`	Configures an interface type and enters interface configuration mode. • In this example, loopback interface 0 is configured.
Step 4	**ip address** *ip-address mask* [**secondary**]	Configures an IP address.

	Command or Action	Purpose
	Example: `Router(config-if)# ip address 172.16.1.1 255.255.255.255`	• In this example, the loopback interface is configured with an IP address of 172.16.1.1.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface loopback1`	Configures an interface type and enters interface configuration mode. • In this example, loopback interface 1 is configured.
Step 7	**ip vrf forwarding** *vrf-name* [**downstream** *vrf-name2*] **Example:** `Router(config-if)# ip vrf forwarding vrf_trans`	Associates a VRF with an interface or subinterface. • In this example, the VRF named vrf_trans is associated with loopback interface 1. **Note** Executing this command on an interface removes the IP address. The IP address should be reconfigured.
Step 8	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.99.1.1 255.255.255.255`	Configures an IP address. • In this example, loopback interface 1 is configured with an IP address of 10.99.1.1.
Step 9	Repeat Step 5 through Step 8 for each VRF to be associated with an interface.	--
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 12	**bgp router-id** {*ip-address*| **vrf auto-assign**} **Example:** `Router(config-router)# bgp router-id vrf auto-assign`	Configures a fixed router ID for the local BGP routing process. • In this example, a BGP router ID is automatically assigned for each VRF instance.

	Command or Action	Purpose
Step 13	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast**router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.
Step 14	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 15	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor is an internal neighbor.
Step 16	**neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 192.168.1.1 update-source loopback0`	Allows BGP sessions to use any operational interface for TCP connections. • In this example, BGP TCP connections for the specified neighbor are sourced with the IP address of the loopback interface rather than the best local address.
Step 17	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. • The example creates a VPNv4 address family session.
Step 18	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:**	Activates the neighbor under the VPNv4 address family. • In this example, the neighbor 172.16.1.1 is activated.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor 172.16.1.1 activate`	
Step 19	**neighbor** {*ip-address*| *peer-group-name*} **send-community**{**both**| **standard**| **extended**} **Example:** `Router(config-router-af)# neighbor 172.16.1.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 172.16.1.1.
Step 20	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 21	**address-family** {**ipv4** [**mdt** | **multicast** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*] | **vpnv4** [**unicast**]} **Example:** `Router(config-router)# address-family ipv4 vrf vrf_trans`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations. • The example specifies that the VRF instance named vrf_trans is to be associated with subsequent IPv4 address family configuration mode commands.
Step 22	**redistribute connected** **Example:** `Router(config-router-af)# redistribute connected`	Redistributes from one routing domain into another routing domain. • In this example, the **connected** keyword is used to represent routes that are established automatically when IP is enabled on an interface. • Only the syntax applicable to this step is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 23	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 40000`	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 192.168.1.1 is an external neighbor.

	Command or Action	Purpose
Step 24	**neighbor** *ip-address* **local-as** *autonomous-system-number* [**no-prepend** [**replace-as** [**dual-as**]]] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 local-as 50000 no-prepend`	Customizes the AS_PATH attribute for routes received from an eBGP neighbor. • The autonomous system number from the local BGP routing process is prepended to all external routes by default. • Use the **no-prepend** keyword to not prepend the local autonomous system number to any routes received from the eBGP neighbor. • In this example, routes from the neighbor at 192.168.1.1 will not contain the local autonomous system number.
Step 25	**neighbor** {*ip-address*| *peer-group-name*} **ebgp-multihop**[*ttl*] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 ebgp-multihop 2`	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. • In this example, BGP is configured to allow connections to or from neighbor 192.168.1.1, which resides on a network that is not directly connected.
Step 26	**neighbor** {*ip-address*| *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Activates the neighbor under the IPV4 address family. • In this example, the neighbor 192.168.1.1 is activated.
Step 27	**neighbor** *ip-address* **allowas-in** [*number*] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 allowas-in 1`	Configures provider edge (PE) routers to allow the readvertisement of all prefixes that contain duplicate autonomous system numbers. • In the example, the PE router with autonomous system number 45000 is configured to allow prefixes from the VRF vrf-trans. The neighboring PE router with the IP address 192.168.1.1 is set to be readvertised once to other PE routers with the same autonomous system number.
Step 28	**no auto-summary** **Example:** `Router(config-router-af)# no auto-summary`	Disables automatic summarization and sends subprefix routing information across classful network boundaries.
Step 29	**no synchronization** **Example:** `Router(config-router-af)# no synchronization`	Enables the Cisco IOS XE software to advertise a network route without waiting for synchronization with an Internal Gateway Protocol (IGP).

	Command or Action	Purpose
Step 30	Repeat Step 20 to Step 29 to configure another VRF instance.	--
Step 31	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 32	**show ip bgp vpnv4** {**all**| **rd** *route-distinguisher*| **vrf** *vrf-name*} **Example:** `Router# show ip bgp vpnv4 all`	(Optional) Displays VPN address information from the BGP table. • In this example, the complete VPNv4 database is displayed. **Note** Only the syntax applicable to this task is used in this example. For more details, see the *Cisco IOS Multiprotocol Label Switching Command Reference*.

Examples

The following sample output assumes that two VRF instances named vrf_trans and vrf_user were configured, each with a separate router ID. The router ID is shown next to the VRF name.

```
Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 172.16.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf vrf_trans) VRF Router ID 10.99.1.2
*> 172.22.0.0       0.0.0.0                  0         32768 ?
r> 172.23.0.0       172.23.1.1               0             0 3 1 ?
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
*> 10.52.1.0/24     172.23.1.1                             0 3 1 ?
*> 10.52.2.1/32     172.23.1.1                             0 3 1 3 i
*> 10.52.3.1/32     172.23.1.1                             0 3 1 3 i
*> 10.99.1.1/32     172.23.1.1               0             0 3 1 ?
*> 10.99.1.2/32     0.0.0.0                  0         32768 ?
Route Distinguisher: 10:1
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
Route Distinguisher: 42:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
r> 172.22.0.0       172.22.1.1               0             0 2 1 ?
*> 172.23.0.0       0.0.0.0                  0         32768 ?
*> 10.21.1.1/32     172.22.1.1                             0 2 1 2 i
*>i10.52.1.0/24     192.168.3.1              0    100      0 ?
*>i10.52.2.1/32     192.168.3.1              0    100      0 3 i
*>i10.52.3.1/32     192.168.3.1              0    100      0 3 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
*> 10.99.1.2/32     172.22.1.1               0             0 2 1 ?
```

Configuration Examples for Per-VRF Assignment of BGP Router ID

Manually Configuring a BGP Router ID per VRF Examples

The following example shows how to configure two VRFs--vrf_trans and vrf_user--with sessions between each other on the same router. The BGP router ID for each VRF is configured manually under separate IPv4 address families. The **show ip bgp vpnv4** command can be used to verify that the router IDs have been configured for each VRF. The configuration starts in global configuration mode.

```
ip vrf vrf_trans
 rd 45000:1
 route-target export 50000:50
 route-target import 40000:1
!
ip vrf vrf_user
 rd 65500:1
 route-target export 65500:1
 route-target import 65500:1
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
router bgp 45000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.3.1 remote-as 45000
 neighbor 192.168.3.1 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.3.1 activate
  neighbor 192.168.3.1 send-community extended
  exit-address-family
 !
 address-family ipv4 vrf vrf_user
  redistribute connected
  neighbor 172.22.1.1 remote-as 40000
  neighbor 172.22.1.1 local-as 50000 no-prepend
  neighbor 172.22.1.1 ebgp-multihop 2
  neighbor 172.22.1.1 activate
  neighbor 172.22.1.1 allowas-in 1
  no auto-summary
  no synchronization
  bgp router-id 10.99.1.1
  exit-address-family
 !
 address-family ipv4 vrf vrf_trans
  redistribute connected
  neighbor 172.23.1.1 remote-as 50000
  neighbor 172.23.1.1 local-as 40000 no-prepend
  neighbor 172.23.1.1 ebgp-multihop 2
  neighbor 172.23.1.1 activate
  neighbor 172.23.1.1 allowas-in 1
  no auto-summary
  no synchronization
  bgp router-id 10.99.1.2
  exit-address-family
```

After the configuration, the output of the **show ip bgp vpnv4 all** command shows the router ID displayed next to the VRF name:

```
Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 45000:1 (default for vrf vrf_trans) VRF Router ID 10.99.1.2
*> 172.22.0.0       0.0.0.0                  0         32768 ?
r> 172.23.0.0       172.23.1.1               0             0 3 1 ?
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
*> 10.52.1.0/24     172.23.1.1                             0 3 1 ?
*> 10.52.2.1/32     172.23.1.1                             0 3 1 3 i
*> 10.52.3.1/32     172.23.1.1                             0 3 1 3 i
*> 10.99.1.1/32     172.23.1.1               0             0 3 1 ?
*> 10.99.2.2/32     0.0.0.0                  0         32768 ?
Route Distinguisher: 50000:1
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
r> 172.22.0.0       172.22.1.1               0             0 2 1 ?
*> 172.23.0.0       0.0.0.0                  0         32768 ?
*> 10.21.1.1/32     172.22.1.1                             0 2 1 2 i
*>i10.52.1.0/24     192.168.3.1              0    100      0 ?
*>i10.52.2.1/32     192.168.3.1              0    100      0 3 i
*>i10.52.3.1/32     192.168.3.1              0    100      0 3 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
*> 10.99.2.2/32     172.22.1.1               0             0 2 1 ?
```

The output of the **show ip bgp vpnv4 vrf** command for a specified VRF displays the router ID in the output header:

```
Router# show ip bgp vpnv4 vrf vrf_user
BGP table version is 43, local router ID is 10.99.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
r> 172.22.0.0       172.22.1.1               0             0 2 1 ?
*> 172.23.0.0       0.0.0.0                  0         32768 ?
*> 10.21.1.1/32     172.22.1.1                             0 2 1 2 i
*>i10.52.1.0/24     192.168.3.1              0    100      0 ?
*>i10.52.2.1/32     192.168.3.1              0    100      0 3 i
*>i10.52.3.1/32     192.168.3.1              0    100      0 3 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
*> 10.99.2.2/32     172.22.1.1               0             0 2 1 ?
```

The output of the **show ip bgp vpnv4 vrf summary** command for a specified VRF displays the router ID in the first line of the output:

```
Router# show ip bgp vpnv4 vrf vrf_user summary
BGP router identifier 10.99.1.1, local AS number 45000
BGP table version is 43, main routing table version 43
8 network entries using 1128 bytes of memory
8 path entries using 544 bytes of memory
16/10 BGP path/bestpath attribute entries using 1856 bytes of memory
6 BGP AS-PATH entries using 144 bytes of memory
3 BGP extended community entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
```

```
BGP using 3744 total bytes of memory
BGP activity 17/0 prefixes, 17/0 paths, scan interval 15 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.22.1.1      4     2      20      21       43    0    0 00:12:33        3
```

When the path is sourced in the VRF, the correct router ID is displayed in the output of the **show ip bgp vpnv4 vrf**command for a specified VRF and network address:

```
Router# show ip bgp vpnv4 vrf vrf_user 172.23.0.0
BGP routing table entry for 65500:1:172.23.0.0/8, version 22
Paths: (1 available, best #1, table vrf_user)
  Advertised to update-groups:
     2          3
  Local
    0.0.0.0 from 0.0.0.0 (10.99.1.1)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
      Extended Community: RT:65500:1
```

Automatically Assigning a BGP Router ID per VRF Examples

The following three configuration examples show different methods of configuring BGP to automatically assign a separate router ID to each VRF instance:

Globally Automatically Assigned Router ID Using Loopback Interface IP Addresses Example

The following example shows how to configure two VRFs--vrf_trans and vrf_user--with sessions between each other on the same router. Under router configuration mode, BGP is globally configured to automatically assign each VRF a BGP router ID. Loopback interfaces are associated with individual VRFs to source an IP address for the router ID. The **show ip bgp vpnv4** command can be used to verify that the router IDs have been configured for each VRF.

```
ip vrf vrf_trans
 rd 45000:1
 route-target export 50000:50
 route-target import 40000:1
!
ip vrf vrf_user
 rd 65500:1
 route-target export 65500:1
 route-target import 65500:1
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface Loopback1
 ip vrf forwarding vrf_user
 ip address 10.99.1.1 255.255.255.255
!
interface Loopback2
 ip vrf forwarding vrf_trans
 ip address 10.99.2.2 255.255.255.255
!
router bgp 45000
 bgp router-id vrf auto-assign
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.3.1 remote-as 45000
 neighbor 192.168.3.1 update-source Loopback0
 !
```

```
address-family vpnv4
  neighbor 192.168.3.1 activate
  neighbor 192.168.3.1 send-community extended
  exit-address-family
 !
 address-family ipv4 vrf vrf_user
  redistribute connected
  neighbor 172.22.1.1 remote-as 40000
  neighbor 172.22.1.1 local-as 50000 no-prepend
  neighbor 172.22.1.1 ebgp-multihop 2
  neighbor 172.22.1.1 activate
  neighbor 172.22.1.1 allowas-in 1
  no auto-summary
  no synchronization
  exit-address-family
 !
 address-family ipv4 vrf vrf_trans
  redistribute connected
  neighbor 172.23.1.1 remote-as 50000
  neighbor 172.23.1.1 local-as 2 no-prepend
  neighbor 172.23.1.1 ebgp-multihop 2
  neighbor 172.23.1.1 activate
  neighbor 172.23.1.1 allowas-in 1
  no auto-summary
  no synchronization
  exit-address-family
```

After the configuration, the output of the **show ip bgp vpnv4 all** command shows the router ID displayed next to the VRF name. Note that the router IDs used in this example are sourced from the IP addresses configured for loopback interface 1 and loopback interface 2. The router IDs are the same as in the Manually Configuring a BGP Router ID per VRF Examples, on page 1194.

```
Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 45000:1 (default for vrf vrf_trans) VRF Router ID 10.99.2.2
*> 172.22.0.0       0.0.0.0                  0         32768 ?
r> 172.23.0.0       172.23.1.1               0             0 3 1 ?
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
*> 10.52.1.0/24     172.23.1.1                            0 3 1 ?
*> 10.52.2.1/32     172.23.1.1                            0 3 1 3 i
*> 10.52.3.1/32     172.23.1.1                            0 3 1 3 i
*> 10.99.1.1/32     172.23.1.1               0             0 3 1 ?
*> 10.99.1.2/32     0.0.0.0                  0         32768 ?
Route Distinguisher: 50000:1
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
r> 172.22.0.0       172.22.1.1               0             0 2 1 ?
*> 172.23.0.0       0.0.0.0                  0         32768 ?
*> 10.21.1.1/32     172.22.1.1                            0 2 1 2 i
*>i10.52.1.0/24     192.168.3.1              0    100      0 ?
*>i10.52.2.1/32     192.168.3.1              0    100      0 3 i
*>i10.52.3.1/32     192.168.3.1              0    100      0 3 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
*> 10.99.1.2/32     172.22.1.1               0             0 2 1 ?
```

Globally Automatically Assigned Router ID with No Default Router ID Example

The following example shows how to configure a router and associate a VRF that is automatically assigned a BGP router ID when no default router ID is allocated.

```
ip vrf vpn1
 rd 45000:1
 route-target export 45000:1
 route-target import 45000:1
!
interface Loopback0
 ip vrf forwarding vpn1
 ip address 10.1.1.1 255.255.255.255
!
router bgp 45000
 bgp router-id vrf auto-assign
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf vpn1
  neighbor 172.22.1.2 remote-as 40000
  neighbor 172.22.1.2 activate
  no auto-summary
  no synchronization
  exit-address-family
```

Assuming that a second router is configured to establish a session between the two routers, the output of the **show ip interface brief** command shows only the VRF interfaces that are configured.

```
Router# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
Serial2/0/0            unassigned      YES NVRAM  administratively down down
Serial3/0/0            unassigned      YES NVRAM  administratively down down
Loopback0              10.1.1.1        YES NVRAM  up                    up
```

The **show ip vrf** command can be used to verify that a router ID is assigned for the VRF:

```
Router# show ip vrf
Name                              Default RD          Interfaces
  vpn1                              45000:1           Loopback0
VRF session is established:
```

Per-VRF Automatically Assigned Router ID Example

The following example shows how to configure two VRFs--vrf_trans and vrf_user--with sessions between each other on the same router. Under the IPv4 address family associated with an individual VRF, BGP is configured to automatically assign a BGP router ID. Loopback interfaces are associated with individual VRFs to source an IP address for the router ID. The output of the **show ip bgp vpnv4** command can be used to verify that the router IDs have been configured for each VRF.

```
ip vrf vrf_trans
 rd 45000:1
 route-target export 50000:50
 route-target import 40000:1
!
ip vrf vrf_user
 rd 65500:1
 route-target export 65500:1
 route-target import 65500:1
```

```
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface Loopback1
 ip vrf forwarding vrf_user
 ip address 10.99.1.1 255.255.255.255
!
interface Loopback2
 ip vrf forwarding vrf_trans
 ip address 10.99.2.2 255.255.255.255
!
router bgp 45000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 192.168.3.1 remote-as 45000
 neighbor 192.168.3.1 update-source Loopback0
 !
address-family vpnv4
  neighbor 192.168.3.1 activate
  neighbor 192.168.3.1 send-community extended
  exit-address-family
!
 address-family ipv4 vrf vrf_user
  redistribute connected
  neighbor 172.22.1.1 remote-as 40000
  neighbor 172.22.1.1 local-as 50000 no-prepend
  neighbor 172.22.1.1 ebgp-multihop 2
  neighbor 172.22.1.1 activate
  neighbor 172.22.1.1 allowas-in 1
  no auto-summary
  no synchronization
  bgp router-id auto-assign
  exit-address-family
 !
 address-family ipv4 vrf vrf_trans
  redistribute connected
  neighbor 172.23.1.1 remote-as 50000
  neighbor 172.23.1.1 local-as 40000 no-prepend
  neighbor 172.23.1.1 ebgp-multihop 2
  neighbor 172.23.1.1 activate
  neighbor 172.23.1.1 allowas-in 1
  no auto-summary
  no synchronization
  bgp router-id auto-assign
  exit-address-family
```

After the configuration, the output of the **show ip bgp vpnv4 all** command shows the router ID displayed next to the VRF name. Note that the router IDs used in this example are sourced from the IP addresses configured for loopback interface 1 and loopback interface 2.

```
Router# show ip bgp vpnv4 all
BGP table version is 43, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 45000:1 (default for vrf vrf_trans) VRF Router ID 10.99.2.2
*> 172.22.0.0       0.0.0.0                  0         32768 ?
r> 172.23.0.0       172.23.1.1               0             0 3 1 ?
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
*> 10.52.1.0/24     172.23.1.1                             0 3 1 ?
*> 10.52.2.1/32     172.23.1.1                             0 3 1 3 i
```

```
*> 10.52.3.1/32     172.23.1.1                             0 3 1 3 i
*> 10.99.1.1/32     172.23.1.1               0             0 3 1 ?
*> 10.99.1.2/32     0.0.0.0                  0         32768 ?
Route Distinguisher: 50000:1
*>i10.21.1.1/32     192.168.3.1              0    100      0 2 i
Route Distinguisher: 65500:1 (default for vrf vrf_user) VRF Router ID 10.99.1.1
r> 172.22.0.0       172.22.1.1               0             0 2 1 ?
*> 172.23.0.0       0.0.0.0                  0         32768 ?
*> 10.21.1.1/32     172.22.1.1                             0 2 1 2 i
*>i10.52.1.0/24     192.168.3.1              0    100      0 ?
*>i10.52.2.1/32     192.168.3.1              0    100      0 3 i
*>i10.52.3.1/32     192.168.3.1              0    100      0 3 i
*> 10.99.1.1/32     0.0.0.0                  0         32768 ?
*> 10.99.1.2/32     172.22.1.1               0             0 2 1 ?
```

Additional References

Related Documents

Related Topic	Document Title
BGP commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*
MPLS commands: complete command syntax, defaults, command mode, command history, usage guidelines, and examples	*Cisco IOS Multiprotocol Label Switching Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

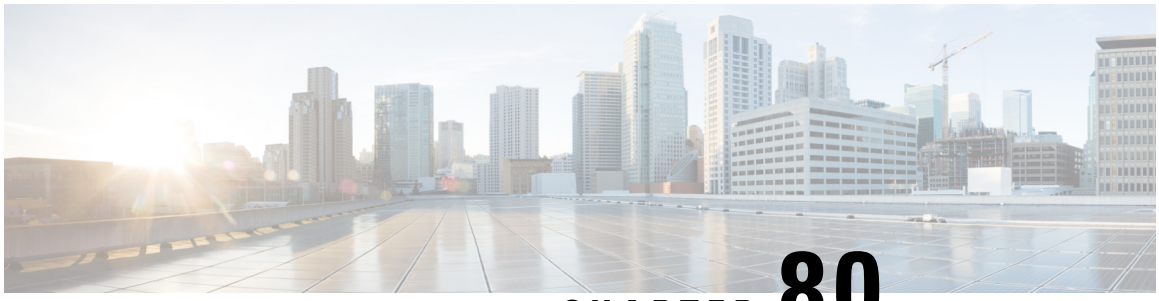
Feature Information for Per-VRF Assignment of BGP Router ID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 105: Feature Information for Per-VRF Assignment of BGP Router ID

Feature Name	Releases	Feature Information
Per-VRF Assignment of BGP Router ID	Cisco IOS XE Release 2.1	The Per-VRF Assignment of BGP Router ID feature introduces the ability to have VRF-to-VRF peering in Border Gateway Protocol (BGP) on the same router. BGP is designed to refuse a session with itself because of the router ID check. The per-VRF assignment feature allows a separate router ID per VRF using a new keyword in the existing **bgp router-id** command. The router ID can be manually configured for each VRF or can be assigned automatically either globally under address family configuration mode or for each VRF. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified by this feature: **bgp router-id**, **show ip bgp vpnv4**.



CHAPTER 80

BGP Next Hop Unchanged

In an external BGP (eBGP) session, by default, the router changes the next hop attribute of a BGP route (to its own address) when the router sends out a route. The BGP Next Hop Unchanged feature allows BGP to send an update to an eBGP multihop peer with the next hop attribute unchanged.

Information About Next Hop Unchanged

BGP Next Hop Unchanged

In an external BGP (eBGP) session, by default, the router changes the next hop attribute of a BGP route (to its own address) when the router sends out a route. If the BGP Next Hop Unchanged feature is configured, BGP will send routes to an eBGP multihop peer without modifying the next hop attribute. The next hop attribute is unchanged.

Note There is an exception to the default behavior of the router changing the next hop attribute of a BGP route when the router sends out a route. When the next hop is in the same subnet as the peering address of the eBGP peer, the next hop is not modified. This is referred to as third party next-hop.

The BGP Next Hop Unchanged feature provides flexibility when designing and migrating networks. It can be used only between eBGP peers configured as multihop. It can be used in a variety of scenarios between two autonomous systems. One scenario is when multiple autonomous systems are connected that share the same IGP, or at least the routers have another way to reach each other's next hops (which is why the next hop can remain unchanged).

A common use of this feature is to configure Multiprotocol Label Switching (MPLS) inter-AS with multihop MP-eBGP for VPNv4 between RRs.

Another common use of this feature is a VPNv4 inter-AS Option C configuration, as defined in RFC4364, Section 10. In this configuration, VPNv4 routes are passed among autonomous systems between RR of

different autonomous systems. The RRs are several hops apart, and have **neighbor next-hop unchanged** configured. PEs of different autonomous systems establish an LSP between them (via a common IGP or by advertising the next-hops--that lead to the PEs--via labeled routes among the ASBRs--routes from different autonomous systems separated by one hop). PEs are able to reach the next hops of the PEs in another AS via the LSPs, and can therefore install the VPNv4 routes in the VRF RIB.

Restriction

The BGP Next Hop Unchanged feature can be configured only between multihop eBGP peers. The following error message will be displayed if you try to configure this feature for a directly connected neighbor:

```
%BGP: Can propagate the nexthop only to multi-hop EBGP neighbor
```

How to Configure BGP Next Hop Unchanged

Configuring the BGP Next Hop Unchanged for an eBGP Peer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family** {**ipv4** | **ipv6** | **l2vpn** | **nsap** | **rtfilter** | **vpnv4** | **vpnv6**}
5. **neighbor** *ip-address* **remote-as** *as-number*
6. **neighbor** *ip-address* **activate**
7. **neighbor** *ip-address* **ebgp-multihop** *ttl*
8. **neighbor** *ip-address* **next-hop-unchanged**
9. **end**
10. **show ip bgp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:**	Enters router configuration mode, and creates a BGP routing process using the specified AS number.

	Command or Action	Purpose
	`Router(config)# router bgp 65535`	
Step 4	**address-family** {**ipv4** | **ipv6** | **l2vpn** | **nsap** | **rtfilter** | **vpnv4** | **vpnv6**} **Example:** `Router(config-router-af)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address family specific configurations.
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.100 remote-as 65600`	Adds an entry to the BGP neighbor table.
Step 6	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.100 activate`	Enables the exchange of information with the peer.
Step 7	**neighbor** *ip-address* **ebgp-multihop** *ttl* **Example:** `Router(config-router-af)# neighbor 10.0.0.100 ebgp-multihop 255`	Configures the local router to accept and initiate connections to external peers that reside on networks that are not directly connected.
Step 8	**neighbor** *ip-address* **next-hop-unchanged** **Example:** `Router(config-router-af)# neighbor 10.0.0.100 next-hop-unchanged`	Configures the router to send BGP updates to the specified BGP peer without modifying the next hop attribute. **Note** This command applies to iBGP routes, not eBGP routes. If you want to apply this setting to both iBGP and eBGP routes, use the **neighbor ip-address next-hopunchanged allpaths** command.
Step 9	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode, and enters privileged EXEC mode.
Step 10	**show ip bgp** **Example:** `Router# show ip bgp`	(Optional) Displays entries in the BGP routing table. • The output will indicate if the **neighbor next-hop-unchanged** command has been configured for the selected address.

Configuring BGP Next Hop Unchanged using Route-Maps

Configuring outbound route-map for eBGP neighbor

To define the route-map and apply outbound policy for neighbor, use **set ip next-hop unchanged** command. This command is applicable for both iBGP and eBGP neighbors and does not require the '**allpaths**' option. The '**allpaths**' is specifically needed in neighbor statement configurations for eBGP to advertise multiple paths.

In the following configuration the next-hop for prefix 1.1.1.1 is not changed while sending to the eBGP neighbor 15.1.1.2:

```
enable
config terminal
router bgp 2
 bgp log-neighbor-changes
 neighbor 15.1.1.2 remote-as 3
 neighbor 15.1.1.2 ebgp-multihop 10
 !
 address-family ipv4
  neighbor 15.1.1.2 activate
  neighbor 15.1.1.2 route-map A out
  exit address-family
!
route-map A permit 10
 match ip address 1
 set ip next-hop unchanged
!
access-list 1 permit 1.1.1.1
end
```

Configuring next-hop unchanged for both iBGP and eBGP path prefixes while sending to eBGP neighbor

To configure next-hop unchanged for both iBGP and eBGP path prefixes while sending to eBGP neighbor, use **next-hop-unchanged allpaths** command.

Note From Cisco IOS XE Denali 16.3 release, the**next-hop-unchanged allpaths** command supports IPv4 and IPv6 address families along with VPNv4 and VPNv6 address families.

In the following configuration the next-hop is not changed for both iBGP and eBGP path prefixes while sending to eBGP neighbor 15.1.1.2:

```
enable
config terminal
router bgp 2
 bgp log-neighbor-changes
 neighbor 15.1.1.2 remote-as 3
 neighbor 15.1.1.2 ebgp-multihop 10
!
address-family ipv4
 neighbor 15.1.1.2 activate
 neighbor 15.1.1.2 next-hop-unchanged allpaths
 exit address-family
!
end
```

Configuration Example for BGP Next Hop Unchanged

Example: BGP Next Hop Unchanged for an eBGP Peer

The following example configures a multihop eBGP peer at 10.0.0.100 in a remote AS. When the local router sends updates to that peer, it will send them without modifying the next hop attribute.

```
router bgp 65535
 address-family ipv4
 neighbor 10.0.0.100 remote-as 65600
 neighbor 10.0.0.100 activate
 neighbor 10.0.0.100 ebgp-multihop 255
 neighbor 10.0.0.100 next-hop-unchanged allpaths
 end
```

Note All address families, such as IPv4, IPv6, VPNv4, VPNv6, L2VPN, and so on support the **next-hop unchanged** command. However, for the address family L2VPN BGP VPLS signaling, you must use the **next-hop self** command for its proper functioning.

Note When configuring eBGP peers for scenarios that require the advertisement of eBGP routes without modifying the next hop attribute, ensure that the '**allpath**' option is included as necessary.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP Outbound Route Map on Route Reflector to Set IP Next Hop for iBGP Peer	“Configuring Internal BGP Features” in the *IP Routing: BGP Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

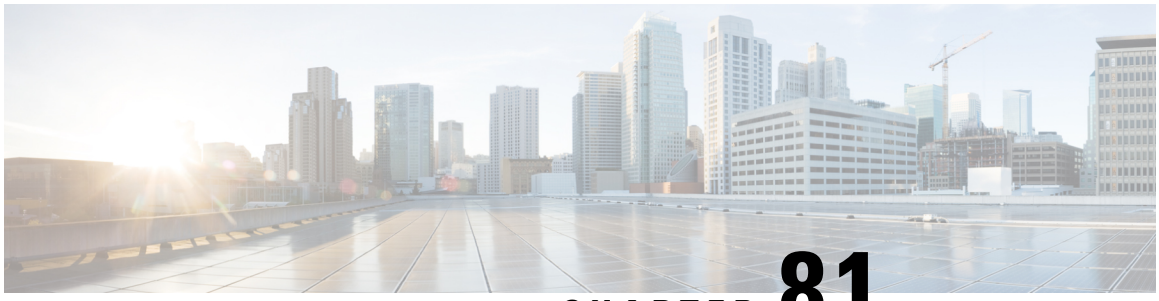
Feature Information for BGP Next Hop Unchanged

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 106: Feature Information for BGP Next Hop Unchanged

Feature Name	Releases	Feature Information
BGP Next Hop Unchanged	Cisco IOS XE Release 2.1	The BGP Next Hop Unchanged feature allows BGP to send an update to an eBGP multihop peer with the next hop attribute unchanged. The following command was added by this feature: **neighbor next-hop-unchanged**.
set ip next-hop unchanged/next-hop-unchanged allpaths IPv4/IPv6	Cisco IOS XE Denali 16.3.1	In Cisco IOS XE Denali 16.3 release, the set ip next-hop unchanged/next-hop-unchanged allpaths IPv4/IPv6 feature extends the support for BGP Next Hop Unchanged to IPv4 and IPv6 allpaths. The set ip next-hop unchanged/next-hop-unchanged allpaths IPv4/IPv6 feature adds two new knobs to support BGP Next Hop Unchanged. The 'set ip next-hop unchanged' knob was added under the route-map and 'next-hop-unchanged allpaths' was added under the neighbor. The following command was modified by this feature: **set ip next-hop unchanged**.



CHAPTER 81

BGP Support for the L2VPN Address Family

BGP support for the Layer 2 Virtual Private Network (L2VPN) address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for BGP Support for the L2VPN Address Family

The BGP Support for L2VPN Address Family feature assumes prior knowledge of Virtual Private Network (VPN), Virtual Private LAN Service (VPLS), and Multiprotocol Layer Switching (MPLS) technologies.

Restrictions for BGP Support for the L2VPN Address Family

- For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration. All other route map commands are supported.
- BGP multipaths and confederations are not supported under the L2VPN address family.

Information About BGP Support for the L2VPN Address Family

L2VPN Address Family

In Cisco IOS XE Release 2.6 and later releases, support for the L2VPN address family is introduced. L2VPN is defined as a secure network that operates inside an unsecured network by using an encryption technology such as IP security (IPsec) or Generic Routing Encapsulation (GRE). The L2VPN address family is configured under BGP routing configuration mode, and within the L2VPN address family the VPLS subsequent address family identifier (SAFI) is supported.

BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. Prefix and path information is stored in the L2VPN database, allowing BGP to make best-path decisions. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the setting up of L2VPN services, which are an integral part of the Cisco IOS Virtual Private LAN Service (VPLS) feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP MPLS network. For more details about VPLS, see the VPLS Autodiscovery: BGP Based feature.

In L2VPN address family, the following BGP commands are supported:

- **bgp nexthop**
- **bgp scan-time**
- **neighbor activate**
- **neighbor advertisement-interval**
- **neighbor allowas-in**
- **neighbor capability**
- **neighbor inherit**
- **neighbor maximum-prefix**
- **neighbor next-hop-self**
- **neighbor next-hop-unchanged**
- **neighbor peer-group**

- **neighbor remove-private-as**
- **neighbor route-map**
- **neighbor route-reflector-client**
- **neighbor send-community**
- **neighbor soft-reconfiguration**
- **neighbor soo**
- **neighbor weight**

Note For route reflectors using L2VPNs, the **neighbor next-hop-self** and **neighbor next-hop-unchanged**commands are not supported.

For route maps used within BGP, all commands related to prefix processing, tag processing, and automated tag processing are ignored when used under L2VPN address family configuration. All other route map commands are supported.

BGP multipaths and confederations are not supported under the L2VPN address family.

VPLS ID

A VPLS ID is a BGP extended community value that identifies the VPLS domain. Manual configuration of this ID is optional because a default VPLS ID is generated using the BGP autonomous system number and the configured VPN ID. A VPLS ID can be composed in one of two ways: with an autonomous system number and an arbitrary number or with an IP address and an arbitrary number.

You can enter a VPLS ID in either of these formats:

- Enter a 16-bit autonomous system number, a colon, and a 32-bit number. For example:

45000:3

- Enter a 32-bit IP address, a colon, and a 16-bit number. For example:

192.168.10.15:1

How to Configure BGP Support for the L2VPN Address Family

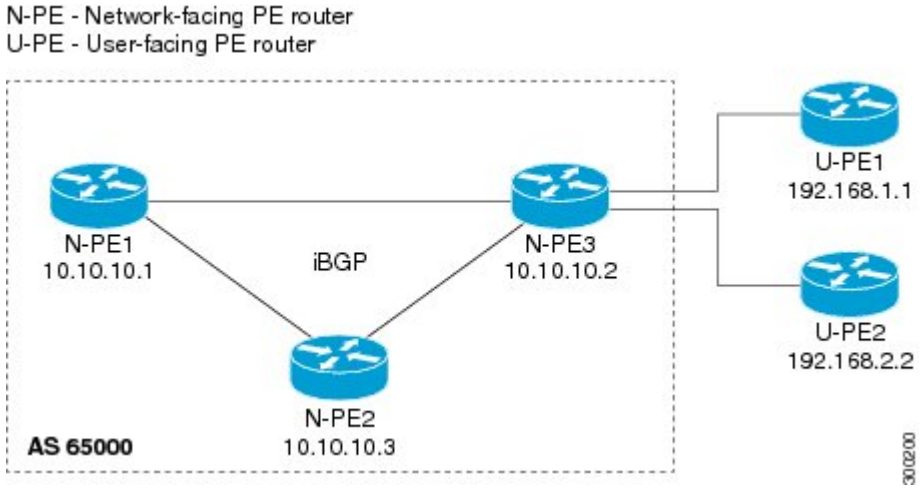
Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family

Perform this task to implement VPLS autodiscovery of each provider edge (PE) router that is a member of a specific VPLS. In Cisco IOS XE Release 2.6, the BGP L2VPN address family was introduced with a separate L2VPN RIB that contains endpoint provisioning information. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time any Layer 2 (L2) virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services.

BGP-based VPLS autodiscovery eliminates the need to manually provision a VPLS neighbor. After a PE router configures itself to be a member of a particular VPLS, information needed to set up connections to remote routers in the same VPLS is distributed by a discovery process. When the discovery process is complete, each member of the VPLS will have the information needed to set up VPLS pseudowires to form the full mesh of pseudowires needed for the VPLS.

This task is configured at router N-PE3 in the figure below and must be repeated at routers N-PE1 and N-PE2 with the appropriate changes such as different IP addresses. For a full configuration of these routers, see the figure below.

Figure 94: Network Diagram for BGP Autodiscovery Using the L2VPN Address Family



In this task, the PE router N-PE3 in the figure above is configured with a Layer 2 router ID, a VPN ID, a VPLS ID, and is enabled to automatically discover other PE routers that are part of the same VPLS domain. A BGP session is created to activate BGP neighbors under the L2VPN address family. Finally, two optional **show** commands are entered to verify the steps in the task.

Before you begin

This task assumes that MPLS is configured with VPLS options. For more details, see the "VPLS Autodiscovery: BGP Based" feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 router-id** *ip-address*
4. **l2 vfi** *vfi-name* **autodiscovery**
5. **vpn id** *vpn-id*
6. **vpls-id** *vpls-id*
7. **exit**
8. Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.
9. **router bgp** *autonomous-system-number*
10. **no bgp default ipv4-unicast**
11. **bgp log-neighbor-changes**
12. **bgp update-delay** *seconds*

13. **neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number*
14. **neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number*
15. Repeat Step 13 and Step 14 to configure other BGP neighbors.
16. **address-family l2vpn** [**vpls**]
17. **neighbor** *ip-address* **activate**
18. **neighbor** {*ip-address*| *peer-group-name*} **send-community**[**both**| **standard**| **extended**]
19. Repeat Step 17 and Step 18 to activate other BGP neighbors under L2VPN address family.
20. **end**
21. **show vfi**
22. **show ip bgp l2vpn vpls** {**all** | **rd** *vpn-rd*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l2 router-id** *ip-address* **Example:** `Router(config)# l2 router-id 10.1.1.3`	Specifies a router ID (in IP address format) for the PE router to use with VPLS autodiscovery pseudowires. • In this example, the L2 router ID is defined as 10.1.1.3.
Step 4	**l2 vfi** *vfi-name* **autodiscovery** **Example:** `Router(config)# l2 vfi customerA autodiscovery`	Creates an L2 VFI, enables the VPLS PE router to automatically discover other PE routers that are part of the same VPLS domain, and enters L2 VFI autodiscovery configuration mode. • In this example, the L2 VFI named customerA is created.
Step 5	**vpn id** *vpn-id* **Example:** `Router(config-vfi)# vpn id 100`	Specifies a VPN ID. • Use the same VPN ID for the PE routers that belong to the same VPN. Make sure that the VPN ID is unique for each VPN in the service provider network. • Use the *vpn-id* argument to specify a number in the range from 1 to 4294967295. • In this example, a VPN ID of 100 is specified.
Step 6	**vpls-id** *vpls-id*	(Optional) Specifies a VPLS ID.

	Command or Action	Purpose
	Example: `Router(config-vfi)# vpls-id 65000:100`	• The VPLS ID is an identifier that is used to identify the VPLS domain. This command is optional because a default VPLS ID is automatically generated using the BGP autonomous system number and the VPN ID configured for the VFI. Only one VPLS ID can be configured per VFI, and the same VPLS ID cannot be configured in multiple VFIs on the same router. • In this example, a VPLS ID of 65000:100 is specified.
Step 7	**exit** **Example:** `Router(config-vfi)# exit`	Exits L2 VFI autodiscovery configuration mode and returns to global configuration mode.
Step 8	Repeat Step 4 through Step 6 to configure other L2 VFIs and associated VPN and VPLS IDs.	--
Step 9	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 10	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	Disables the IPv4 unicast address family for the BGP routing process. **Note** Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the **neighbor remote-as** router configuration command unless you configure the **no bgp default ipv4-unicast**router configuration command before configuring the **neighbor remote-as** command. Existing neighbor configurations are not affected.
Step 11	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 12	**bgp update-delay** *seconds* **Example:** `Router(config-router)# bgp update-delay 1`	Sets the maximum initial delay period before a BGP-speaking networking device sends its first updates. • Use the *seconds* argument to set the delay period.
Step 13	**neighbor** {*ip-address*| *peer-group-name*} **remote-as** *autonomous-system-number* **Example:**	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.

	Command or Action	Purpose
	`Router(config-router)# neighbor 10.10.10.1 remote-as 65000`	• If the *autonomous-system-number* argument matches the autonomous system number specified in the **router bgp** command, the neighbor is an internal neighbor. • If the *autonomous-system-number* argument does not match the autonomous system number specified in the **router bgp** command, the neighbor is an external neighbor. • In this example, the neighbor at 10.10.10.1 is an internal BGP neighbor.
Step 14	**neighbor** {*ip-address*| *peer-group-name*} **update-source** *interface-type interface-number* **Example:** `Router(config-router)# neighbor 10.10.10.1 update-source loopback 1`	(Optional) Configures a router to select a specific source or interface to receive routing table updates. • This example uses a loopback interface. The advantage to this configuration is that the loopback interface is not as susceptible to the effects of a flapping interface.
Step 15	Repeat Step 13 and Step 14 to configure other BGP neighbors.	--
Step 16	**address-family l2vpn** [**vpls**] **Example:** `Router(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The optional **vpls** keyword specifies that VPLS endpoint provisioning information is to be distributed to BGP peers. • In this example, an L2VPN VPLS address family session is created.
Step 17	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 10.10.10.1 activate`	Enables the neighbor to exchange information for the L2VPN VPLS address family with the local router. **Note** If you have configured a BGP peer group as a neighbor, you do not use this step. BGP peer groups are activated when a BGP parameter is configured. For example, the **neighbor send-community** command in the next step will automatically activate a peer group.
Step 18	**neighbor** {*ip-address*| *peer-group-name*} **send-community**[**both**| **standard**| **extended**] **Example:** `Router(config-router-af)# neighbor 10.10.10.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.

	Command or Action	Purpose
Step 19	Repeat Step 17 and Step 18 to activate other BGP neighbors under L2VPN address family.	--
Step 20	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 21	**show vfi** **Example:** `Router# show vfi`	(Optional) Displays information about the configured VFI instances.
Step 22	**show ip bgp l2vpn vpls** {**all** | **rd** *vpn-rd*} **Example:** `Router# show ip bgp l2vpn vpls all`	(Optional) Displays information about the L2 VPN VPLS address family.

Examples

The following is sample output from the **show vfi** command that shows two VFIs, CustomerA and CustomerB, with their associated VPN and VPLS IDs:

```
Router# show vfi
Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No
VFI name: customerA, state: down, type: multipoint
  VPN ID: 100, VPLS-ID: 65000:100
  RD: 65000:100, RT: 65000:100
  Local attachment circuits:
  Neighbors connected via pseudowires:
  Peer Address     VC ID        Discovered Router ID    S
  10.10.10.1         100        10.10.10.99             Y
VFI name: customerB, state: down, type: multipoint
  VPN ID: 200, VPLS-ID: 65000:200
  RD: 65000:200, RT: 65000:200
  Local attachment circuits:
  Neighbors connected via pseudowires:
  Peer Address     VC ID        Discovered Router ID    S
  10.10.10.3         200                10.10.10.98    Y
```

The following is sample output from the **show ip bgp l2vpn vpls all** command that shows two VFIs identified by their VPN route distinguisher:

```
Router# show ip bgp l2vpn vpls all
BGP table version is 5, local router ID is 10.10.10.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 65000:100
*> 65000:100:10.10.10.1/96
                    0.0.0.0                              32768 ?
*>i65000:100:192.168.1.1/96
```

```
                    10.10.10.2               0    100      0 ?
Route Distinguisher: 65000:200
*> 65000:200:10.10.10.3/96
                    0.0.0.0                            32768 ?
*>i65000:200:192.168.2.2/96
                    10.10.10.2               0    100      0 ?
```

What to Do Next

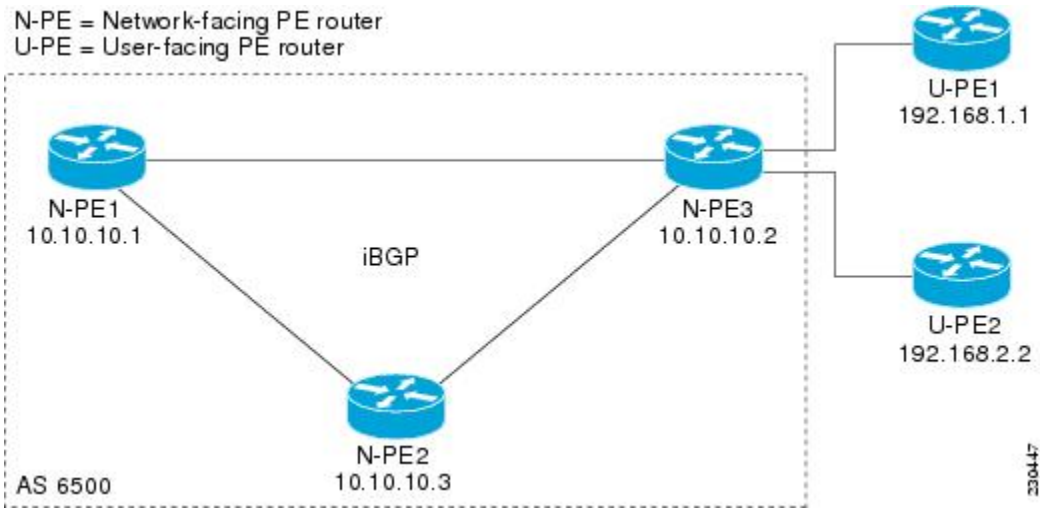
To configure more VPLS features, see the main VPLS documentation in the "VPLS Autodiscovery: BGP Based" module in the *MPLS Layer 2 VPNs Configuration Guide*.

Configuration Examples for BGP Support for the L2VPN Address Family

Example: Configuring VPLS Autodiscovery Using BGP and the L2VPN Address Family

In this configuration example, all the routers in autonomous system 65000 in the figure below are configured to provide BGP support for the L2VPN address family. VPLS autodiscovery is enabled and L2 VFI and VPN IDs are configured. BGP neighbors are configured and activated under L2VPN address family to ensure that the VPLS endpoint provisioning information is saved to a separate L2VPN RIB and then distributed to the other BGP peers in BGP update messages. When the endpoint information is received by the BGP peers, a pseudowire mesh is set up to support L2VPN-based services.

Figure 95: Network Diagram for VPLS Autodiscovery Using BGP and the L2VPN Address Family



Router N-PE1

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
```

```
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 1000 2000
mpls label protocol ldp
l2 router-id 10.1.1.1
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.1 255.255.255.255
!
interface GigabitEthernet0/0/1
 description Backbone interface
 ip address 10.0.0.1 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.10.10.2 remote-as 65000
 neighbor 10.10.10.2 update-source Loopback 1
 neighbor 10.10.10.3 remote-as 65000
 neighbor 10.10.10.3 update-source Loopback 1
!
 address-family l2vpn vpls
 neighbor 10.10.10.2 activate
 neighbor 10.10.10.2 send-community extended
 neighbor 10.10.10.3 activate
 neighbor 10.10.10.3 send-community extended
 exit-address-family
!
ip classless
```

Router N-PE2

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.2
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.2 255.255.255.255
!
interface GigabitEthernet0/0/1
 description Backbone interface
```

```
 ip address 10.0.0.2 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.10.10.1 remote-as 65000
 neighbor 10.10.10.1 update-source Loopback1
 neighbor 10.10.10.3 remote-as 65000
 neighbor 10.10.10.3 update-source Loopback1
!
 address-family l2vpn vpls
 neighbor 10.10.10.1 activate
 neighbor 10.10.10.1 send-community extended
 neighbor 10.10.10.3 activate
 neighbor 10.10.10.3 send-community extended
 exit-address-family
!
ip classless
```

Router N-PE3

```
ip subnet-zero
ip cef
no ip dhcp use vrf connected
!
no mpls traffic-eng auto-bw timers frequency 0
mpls label range 2000 3000
mpls label protocol ldp
l2 router-id 10.1.1.3
l2 vfi auto autodiscovery
 vpn id 100
!
pseudowire-class mpls
 encapsulation mpls
!
interface Loopback1
 ip address 10.1.1.3 255.255.255.255
!
interface GigabitEthernet0/0/1
 description Backbone interface
 ip address 10.0.0.3 255.255.255.0
 mpls ip
!
router ospf 1
 log-adjacency-changes
 network 10.10.1.0 0.0.0.255 area 0
 network 192.168.0.0 0.0.0.255 area 0
!
router bgp 65000
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp update-delay 1
 neighbor 10.10.10.1 remote-as 65000
 neighbor 10.10.10.1 update-source Loopback1
 neighbor 10.10.10.2 remote-as 65000
 neighbor 10.10.10.2 update-source Loopback1
```

```
!
 address-family l2vpn vpls
 neighbor 10.10.10.1 activate
 neighbor 10.10.10.1 send-community extended
 neighbor 10.10.10.2 activate
 neighbor 10.10.10.2 send-community extended
 exit-address-family
!
ip classless
```

Where to Go Next

For more details about configuring VPLS autodiscovery, see the "VPLS Autodiscovery: BGP Based" module in the *MPLS Layer 2 VPNs Configuration Guide*.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

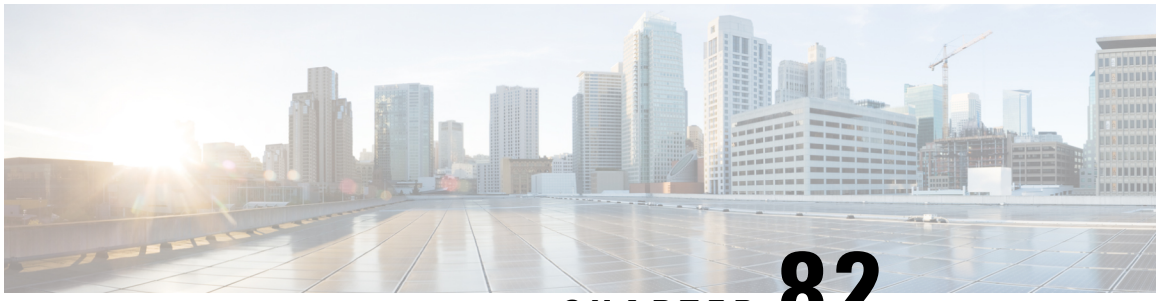
Feature Information for BGP Support for the L2VPN Address Family

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 107: Feature Information for BGP Support for the L2VPN Address Family

Feature Name	Releases	Feature Information
BGP Support for the L2VPN Address Family	Cisco IOS XE Release 2.6 Cisco IOS XE Release 3.3SG	BGP support for the L2VPN address family introduces a BGP-based autodiscovery mechanism to distribute L2VPN endpoint provisioning information. BGP uses a separate L2VPN Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 VFI is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to set up a pseudowire mesh to support L2VPN-based services. The following commands were introduced or modified by this feature: **address-family l2vpn**, **clear ip bgp l2vpn**, and **show ip bgp l2vpn**.



CHAPTER 82

BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced.

Prerequisites for BGP Event-Based VPN Import

Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled on all participating routers.

Information About BGP Event-Based VPN Import

BGP Event-Based VPN Import

The BGP Event-Based VPN Import feature introduces a modification to the existing BGP path import process. BGP Virtual Private Network (VPN) import provides importing functionality for BGP paths where BGP paths are imported from the BGP VPN table into a BGP virtual routing and forwarding (VRF) topology. In the existing path import process, when path updates occur, the import updates are processed during the next scan time which is a configurable interval of 5 to 15 seconds. The scan time adds a delay in the propagation of routes. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available.

Using the BGP Event-Based VPN Import feature, convergence times are significantly reduced because provider edge (PE) routers can propagate VPN paths to customer edge (CE) routers without the scan time delay.

Configuration changes such as adding imported route-targets to a VRF are not processed immediately, and are still handled during the 60-second periodic scanner pass.

Import Path Selection Policy

Event-based VPN import introduces three path selection policies:

- All—Import all available paths from the exporting net that match any route target (RT) associated with the importing VRF instance.
- Best path—Import the best available path that matches the RT of the VRF instance. If the best path in the exporting net does not match the RT of the VRF instance, a best available path that matches the RT of the VRF instance is imported.
- Multipath—Import the best path and all paths marked as multipaths that match the RT of the VRF instance. If there are no best path or multipath matches, then the best available path is selected.

Multipath and best path options can be restricted using an optional keyword to ensure that the selection is made only on the configured option. If the **strict** keyword is configured in the **import path selection** command, the software disables the fall back safety option of choosing the best available path. If no paths appropriate to the configured option (best path or multipath) in the exporting net match the RT of the VRF instance, then no paths are imported. This behavior matches the behavior of the software before the BGP Event-Based VPN Import feature was introduced.

When the restriction is not set, paths that are imported as the best available path are tagged. In **show** command output these paths are identified with the wording, "imported safety path."

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as "not-in-vrf" in the **show** command output. Any path that is marked as "not-in-vrf" is not considered as a best path because paths not in the VRF appear less attractive than paths in the VRF.

Import Path Limit

To control the memory utilization, a maximum limit of the number of paths imported from an exporting net can be specified per importing net. When a selection is made of paths to be imported from one or more exporting net, the first selection priority is a best path, the next selection priority is for multipaths, and the lowest selection priority is for nonmultipaths.

How to Configure BGP Event-Based VPN Import

Configuring a Multiprotocol VRF

Perform this task to configure a multiprotocol VRF that allows you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. In this task, only the IPv4 address family is configured, but we recommend using the multiprotocol VRF configuration for all new VRF configurations.

Note This task is not specific to the BGP Event-Based VPN Import feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **address-family ipv4** [**unicast**]
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **no shutdown**
13. **exit**
14. Repeat Step 3 through Step 13 to bind other VRF instances with an interface.
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf-A`	Configures a VRF routing table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 45000:1`	Creates routing and forwarding tables and specifies the default route distinguisher for a VPN. • Use the *route-distinguisher* argument to add an 8-byte value to an IPv4 prefix to create a unique VPN IPv4 prefix.

	Command or Action	Purpose
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target both 45000:100`	Creates a route target extended community for a VRF. • Use the **import** keyword to import routing information from the target VPN extended community. • Use the **export** keyword to export routing information to the target VPN extended community. • Use the **both** keyword to both import routing information from, and export routing information to, the target VPN extended community. • Use the *route-target-ext-community* argument to add the route target extended community attributes to the VRF's list of import, export, or both (import and export) route target extended communities.
Step 6	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters VRF address family configuration mode. • This step is required here to specify an address family for the VRF defined in the previous steps.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 1/1`	Enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding vrf-A`	Associates a VRF instance with the interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:**	Configures an IP address for the interface.

	Command or Action	Purpose
	`Router(config-if)# ip address 10.4.8.149 255.255.255.0`	
Step 12	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Restarts a disabled interface.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 14	Repeat Step 3 through Step 13 to bind other VRF instances with an interface.	--
Step 15	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Event-Based VPN Import Processing for BGP Paths

Perform this task to reduce convergence times when BGP paths change by configuring event-based processing for importing BGP paths into a VRF table. Two new CLI commands allow the configuration of a maximum number of import paths per importing net and the configuration of a path selection policy.

Before you begin

This task assumes that you have previously configured the VRF to be used with the VRF address family syntax. To configure a VRF, see the "Configuring a Multiprotocol VRF" section earlier in this module.

Complete BGP neighbor configuration is also assumed. For an example configuration, see the "Example: Configuring Event-Based VPN Import Processing for BGP Paths" section in this module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **import path selection** {**all** | **bestpath** [**strict**] | **multipath** [**strict**]}
6. **import path limit** *number-of-import-paths*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf vrf-A`	Specifies the IPv4 address family and enters address family configuration mode. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**import path selection** {**all** | **bestpath** [**strict**] | **multipath** [**strict**]} **Example:** `Router(config-router-af)# import path selection all`	Specifies the BGP path selection policy for importing routes into a VRF table. • In this example, all paths that match any RT of the VRF instance are imported.
Step 6	**import path limit** *number-of-import-paths* **Example:** `Router(config-router-af)# import path limit 3`	Specifies, per importing net, a maximum number of BGP paths that can be imported from an exporting net.
Step 7	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Monitoring and Troubleshooting BGP Event-Based VPN Import Processing

Perform the steps in this task as required to monitor and troubleshoot the BGP event-based VPN import processing.

Only partial command syntax for the **show** commands used in this task is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*network-address* [*mask*]]
3. **show ip route** [**vrf** *vrf-name*] [*ip-address* [*mask*]]
4. **debug ip bgp vpnv4 unicast import** {**events** | **updates** [*access-list*]}

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 show ip bgp vpnv4 {all | rd route-distinguisher | vrf vrf-name} [network-address [mask]]

In this example output, a safe import path selection policy is in effect because the **strict** keyword is not configured using the **import path selection** command. When a path is imported as the best available path (when the bestpath or multipaths are not eligible for import), the path is marked with "imported safety path," as shown in the output.

Example:

```
Router# show ip bgp vpnv4 all 172.17.0.0

BGP routing table entry for 45000:1:172.17.0.0/16, version 10
Paths: (1 available, best #1, table vrf-A)
Flag: 0x820
   Not advertised to any peer
   2, imported safety path from 50000:2:172.17.0.0/16
     10.0.101.1 from 10.0.101.1 (10.0.101.1)
       Origin IGP, metric 200, localpref 100, valid, internal, best
       Extended Community: RT:45000:100
```

The paths existing in an exporting net that are considered for import into a VRF instance may have been received from another peer router and were not subject to the VPN importing rules. These paths may contain the same route-distinguisher (RD) information because the RD information is local to a router, but some of these paths do not match the RT of the importing VRF instance and are marked as "not-in-vrf" in the **show** command output.

In the following example output, a path was received from another peer router and was not subject to the VPN importing rules. This path, 10.0.101.2, was added to the VPNv4 table and associated with the vrf-A net because it contains a match of the RD information although the RD information was from the original router. This path is not, however, an RT match for vrf-A and is marked as "not-in-vrf." Note that on the net for vrf-A, this path is not the bestpath because any paths that are not in the VRF appear less attractive than paths in the VRF.

Example:

```
Router# show ip bgp vpnv4 all 172.17.0.0

BBGP routing table entry for 45000:1:172.17.0.0/16, version 11
Paths: (2 available, best #2, table vrf-A)
```

```
Flag: 0x820
   Not advertised to any peer
   2
     10.0.101.2 from 10.0.101.2 (10.0.101.2)
       Origin IGP, metric 100, localpref 100, valid, internal, not-in-vrf
       Extended Community: RT:45000:200
       mpls labels in/out nolabel/16
   2
     10.0.101.1 from 10.0.101.1 (10.0.101.1)
       Origin IGP, metric 50, localpref 100, valid, internal, best
       Extended Community: RT:45000:100
       mpls labels in/out nolabel/16
```

Step 3 **show ip route** [**vrf** *vrf-name*] [*ip-address* [*mask*]]

In this example output, information about the routing table for VRF vrf-A is displayed:

Example:

```
Router# show ip route vrf vrf-A 172.17.0.0

Routing Table: vrf-A
Routing entry for 172.17.0.0/16
  Known via "bgp 1", distance 200, metric 50
  Tag 2, type internal
  Last update from 10.0.101.33 00:00:32 ago
  Routing Descriptor Blocks:
  * 10.0.101.33 (default), from 10.0.101.33, 00:00:32 ago
      Route metric is 50, traffic share count is 1
      AS Hops 1
      Route tag 2
      MPLS label: 16
      MPLS Flags: MPLS Required
```

Step 4 **debug ip bgp vpnv4 unicast import** {**events** | **updates** [*access-list*]}

Use this command to display debugging information related to the importing of BGP paths into a VRF instance table. The actual output depends on the commands that are subsequently entered.

Note If no access list to filter prefixes is specified when using the updates keyword, all updates for all prefixes are displayed and this may slow down your network.

Example:

```
Router# debug ip bgp vpnv4 unicast import events

BGP import events debugging is on
```

Configuration Examples for BGP Event-Based VPN Import

Example: Configuring Event-Based VPN Import Processing for BGP Paths

In this example, a VRF (vrf-A) is configured and VRF forwarding is applied to Fast Ethernet interface 1/1. In address family mode, the import path selection is set to all and the number of import paths is set to 3. Two BGP neighbors are configured under the IPv4 address family and activated under the VPNv4 address family.

```
vrf definition vrf-A
 rd 45000:1
 route-target import 45000:100
 address-family ipv4
  exit-address-family
!
interface FastEthernet1/1
 no ip address
 vrf forwarding vrf-A
 ip address 10.4.8.149 255.255.255.0
 no shut
 exit
!
router bgp 45000
 network 172.17.1.0 mask 255.255.255.0
 neighbor 192.168.1.2 remote-as 40000
 neighbor 192.168.3.2 remote-as 50000
 address-family ipv4 vrf vrf-A
  import path selection all
  import path limit 3
  exit-address-family
 address-family vpnv4
  neighbor 192.168.1.2 activate
  neighbor 192.168.3.2 activate
  end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

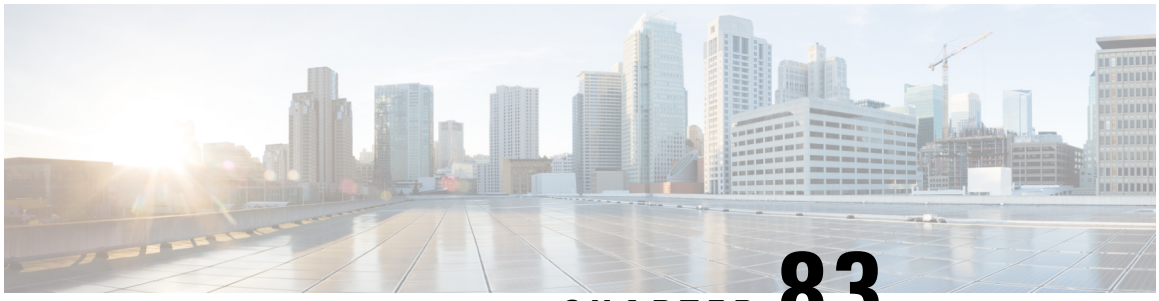
Feature Information for BGP Event-Based VPN Import

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 108: Feature Information for BGP Event-Based VPN Import

Feature Name	Releases	Feature Information
BGP Event-Based VPN Import	Cisco IOS XE Release 2.6 Cisco IOS XE Release 3.3SG	The BGP Event-Based VPN Import feature introduces a modification to the existing Border Gateway Protocol (BGP) path import process. The enhanced BGP path import is driven by events; when a BGP path changes, all of its imported copies are updated as soon as processing is available. Convergence times are significantly reduced because there is no longer any delay in the propagation of routes due to the software waiting for a periodic scanner time interval before processing the updates. To implement the new processing, new command-line interface (CLI) commands are introduced. The following commands were introduced or modified: • **bgp scan-time** • **import path limit** • **import path selection** • **maximum-path ebgp** • **maximum-path ibgp** • **show ip bgp vpnv4** • **show ip bgp vpnv6**



CHAPTER 83

BGP Best External

The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. The BGP Best External feature advertises the most preferred route among those received from external neighbors as a backup route. This feature is beneficial in active-backup topologies, where service providers use routing policies that cause a border router to choose a path received over an Interior Border Gateway Protocol (iBGP) session (of another border router) as the best path for a prefix even if it has an Exterior Border Gateway Protocol (eBGP) learned path. This active-backup topology defines one exit or egress point for the prefix in the autonomous system and uses the other points as backups if the primary link or eBGP peering is unavailable. The policy causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because it does not advertise any path for such prefixes. To cope with this situation, some devices advertise one externally learned path called the best external path.

Prerequisites for BGP Best External

- The Bidirectional Forwarding Detection (BFD) protocol must be enabled to quickly detect link failures.
- Ensure that the BGP and theMultiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- The backup path must have a unique next hop that is not the same as the next hop of the best path.
- BGP must support lossless switchover between operational paths.

Restrictions for BGP Best External

- The BGP Best External feature will not install a backup path if BGP Multipath is installed and a multipath exists in the BGP table. One of the multipaths automatically acts as a backup for the other paths.
- The BGP Best External feature is not supported with the following features:

  - MPLS VPN Carrier Supporting Carrier
  - MPLS VPN Per Virtual Routing and Forwarding (VRF) Label

- The BGP Best External feature cannot be configured with Multicast or L2VPN VRF address families.
- The BGP Best External feature cannot be configured on a route reflector, unless it is running Cisco IOS XE Release 3.4S or later.
- The BGP Best External feature does not support NSF/SSO. However, ISSU is supported if both Route Processors have the BGP Best External feature configured.
- The BGP Best External feature can only be configured on VPNv4, VPNv6, IPv4 VRF, and IPv6 VRF address families.
- When you configure the BGP Best External feature using the **bgp advertise-best-external** command, you need not enable the BGP PIC feature with the **bgp additional-paths install** command. The BGP PIC feature is automatically enabled by the BGP Best External feature.
- When you configure the BGP Best External feature, it will override the functionality of the "MPLS VPN--BGP Local Convergence" feature. However, you do not have to remove the **protection local-prefixes** command from the configuration.
- BGP Best External Path with MPLS VPN Inter-AS Option C is supported with deployments that use IPv4 addresses and labels. The configuration is not supported with IPv6 addresses and labels.

Information About BGP Best External

BGP Best External Overview

Service providers use routing policies that cause a border router to choose a path received over an iBGP session (of another border router) as the best path for a prefix even if it has an eBGP learned path. This practice is popularly known as active-backup topology and is done to define one exit or egress point for the prefix in the autonomous system and to use the other points as backups if the primary link or eBGP peering is unavailable.

The policy, though beneficial, causes the border router to hide the paths learned over its eBGP sessions from the autonomous system because the border router does not advertise any path for such prefixes. To cope with this situation, some routers advertise one externally learned path called the best external path. The best external behavior causes the BGP selection process to select two paths to every destination:

- The best path is selected from the complete set of routes known to that destination.
- The best external path is selected from the set of routes received from its external peers.

BGP advertises the best path to external peers. Instead of withdrawing the best path from its internal peers when it selects an iBGP path as the best path, BGP advertises the best external path to the internal peers.

The BGP Best External feature is an essential component of the Prefix-Independent Convergence (PIC) edge for both Internet access and MPLS VPN scenarios and makes alternate paths available in the network in the active-backup topology.

What the Best External Route Means

The BGP Best External feature uses a “best external route” as a backup path, which, according to draft-marques-idr-best-external, is the most preferred route among those received from external neighbors. The most preferred route from external neighbors can be the following:

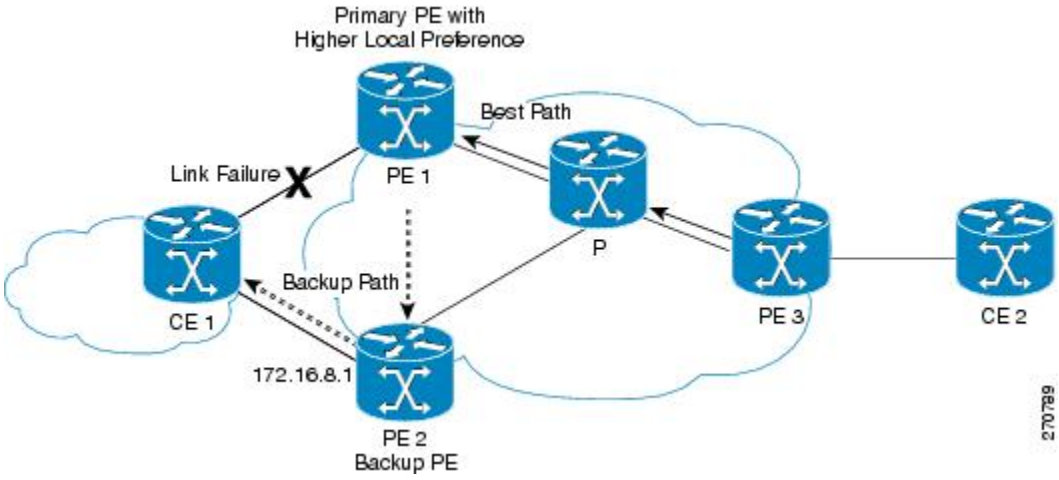
- Two routers in different clusters that have an Interior Border Gateway Protocol (iBGP) session between them.
- Two routers in different autonomous systems of a confederation that have an External Border Gateway Protocol (eBGP) session between them.

The best external route might be different from the best route installed in the Routing Information Base (RIB). The best route could be an internal route. By allowing the best external route to be advertised and stored, in addition to the best route, networks gain faster restoration of connectivity by providing additional paths that may be used if the primary path fails.

How the BGP Best External Feature Works

The BGP Best External feature is based on Internet Engineering Task Force (IETF) draft-marques-idr-best-external.txt. The BGP Best External feature advertises a best external route to its internal peers as a backup route. The backup route is stored in the RIB and Cisco Express Forwarding. If the primary path fails, the BGP PIC functionality enables the best external path to take over, enabling faster restoration of connectivity.

Figure 96: MPLS VPN: Best External at the Edge of MPLS VPN



The figure above shows an MPLS VPN using the BGP Best External feature. The network includes the following components:

- eBGP sessions exist between the provider edge (PE) and customer edge (CE) routers.
- PE1 is the primary router and has a higher local preference setting.
- Traffic from CE2 uses PE1 to reach router CE1.
- PE1 has two paths to reach CE1.
- CE1 is dual-homed with PE1 and PE2.

- PE1 is the primary path and PE2 is the backup path.

In the figure above, traffic in the MPLS cloud flows through PE1 to reach CE1. Therefore, PE2 uses PE1 as the best path and PE2 as the backup path.

PE1 and PE2 are configured with the BGP Best External feature. BGP computes both the best path (the PE1-CE1 link) and a backup path (PE2) and installs both paths into the RIB and Cisco Express Forwarding. The best external path (PE2) is advertised to the peer routers, in addition to the best path.

When Cisco Express Forwarding detects a link failure on the PE1-CE1 link, Cisco Express Forwarding immediately switches to the backup path PE2. Traffic is quickly rerouted due to local Fast Convergence in Cisco Express Forwarding using the backup path. Thus, traffic loss is minimized and fast convergence is achieved.

Configuration Modes for Enabling BGP Best External

You can enable the BGP Best External feature in different modes, each of which protects Virtual Routing and Forwarding (VRF) in its own way:

- If you issue the **bgp advertise-best-external** command in VPNv4 address family configuration mode, it applies to all IPv4 VRFs. If you issue the command in this mode, you need not issue it for specific VRFs.
- If you issue the **bgp advertise-best-external** command in IPv4 address family configuration mode, it applies only to that VRF.

BGP Best External Path on RR for Intercluster

Beginning with Cisco IOS XE Release 3.4S, BGP Best External is extended to BGP Best External for Intercluster RRs. This feature provides path diversity between RR clusters, providing best external functionality toward non-client iBGP peers. The feature is also known as the “intercluster best external path.”

Best external path at an RR means the best path within the RR’s cluster. This path might also be referred to as the best internal path.

When an RR (RR1) chooses a non-client iBGP path (that is, a path learned from another RR, let’s say RR2) as its overall best, with the BGP Best External for Intercluster RRs feature, RR1 will be able to advertise its best internal path to the non-client iBGP peers. This will help RR2 to learn an additional path, providing a diverse path.

Best external functionality at RRs is only for non-client iBGP peers. An RR cannot advertise best external paths to its clients because it has to advertise its overall bestpath (which can be either a client path or non-client or eBGP path).

The best external path calculated by the RR is the best internal path for the cluster. It will be advertised to the non-client iBGP peers only when the overall best path at this RR is a non-client iBGP path.

When there are multiple RRs, each in its own cluster, each RR must have the **neighbor advertise best-external** command configured for each of its neighbor RRs.

If the RR is in the forwarding plane, the **bgp additional paths install** command is necessary.

CLI Differences for Best External Path on an RR for Intercluster

Prior to Cisco IOS XE Release 3.4S, the BGP Best External feature was allowed on a PE only, and it was configured by the **bgp advertise-best-external** command. The calculation of the backup path, installation, and advertisement were tied together in one command.

Beginning with Cisco IOS XE Release 3.4S, the BGP Best External feature is allowed on PEs and RRs. The functionality of the **bgp advertise-best-external** command is divided among the following three commands that calculate, install, and advertise the best external path:

- **bgp additional-path select best-external**
- **bgp additional-path install**
- **neighbor advertise diverse-path best-external**

If the **bgp additional-path select best-external** command is not configured, the system will calculate and install the best external path, but not advertise it.

The **neighbor advertise diverse-path best-external** command enables the advertisement of the best external path to the specified neighbor.

Rules Used to Calculate the BGP Best External Path for Intercluster RRs

The best internal path implementation on an RR toward non-clients (different cluster RRs) is calculated based on the following rules:

1. Calculate the overall primary bestpath on the RR per the normal bestpath selection rules.
2. If a backup path configuration is enabled, calculate the second bestpath (which is a different path from the primary bestpath selected in Rule 1 and has a different nexthop from this bestpath), which is marked as the backup path. Backup path selection is enabled using the **bgp additional-paths install** or **bgp additional-paths select [best-external] [backup]** command.
3. If the overall best path on the RR is a non-client iBGP path and not an eBGP path, calculate the best external/internal path from the remaining paths after excluding results from Rule 1 and Rule 2 and by ignoring all the other paths from the other clusters and run normal bestpath rules by including all the remaining eBGP and iBGP paths. Select the newly obtained bestpath and mark it as the best internal path.
4. Advertise this best internal path, which is either eBGP (received from CE peers for RR/ASBR) or iBGP (received from RR clients) toward the non-client RRs when **neighbor advertise best-external** is configured towards the non-client RRs.
5. If the overall bestpath is a path received from either an RR client or eBGP peer (in case of RR/ASBR) either an iBGP or an eBGP path will be chosen as bestpath per the normal bestpath algorithm. Because the overall bestpath is an internal client path, the normal advertisement rules will automatically advertise this path to non-client iBGP peers/RRs. This behavior is the same as the existing behavior (when best external is not enabled on RRs) when an RR client's path is chosen as the overall bestpath.
6. We do not allow a best external path to be configured on an RR towards RR-clients. The **neighbor advertise best-external** command can be configured on RR/ASBR only for non-clients or peering with RRs in the other clusters.
7. When multipath is enabled on the RR and only when the overall bestpath is from a non-client and if some of the intracluster client paths are also marked as multipaths, when best external is enabled on the RR

(**neighbor advertise best-external** towards the RR non-client), the algorithm selects the older multipath among the intra-cluster client multipaths (paths obtained from RR clients and eBGP peers within the cluster) and marks it as best internal path and announces it to the non-clients as best external path, so that the non-clients get path diversity from this cluster. If there are no intra-cluster multipaths found, we choose the best external path per Rules 3 through 5.

BGP Best External Path with MPLS Inter-AS Options B and C

BGP Best External Path with MPLS VPN Inter-AS Option B

With this feature, you can configure the border routers in an MPLS VPN Inter-AS Option B deployment to compute and advertise a best external path based on routes received from EBGP peers. The border routers advertise the best external path so computed to Internal BGP (IBGP) peers. The best external path is advertised in addition to the best path and serves as a back-up path. If a link in the best path fails, traffic flows along the best external path.

Consider the MPLS VPN Inter-AS Option B deployment shown in the following diagram:

Figure 97: MPLS VPN Inter-AS Option B Deployment

ASBR1 and ASBR2 exchange VPNv4/VPNv6 addresses, PE node loopback addresses, and labels using External BGP (EBGP). Similarly, ASBR3 and ASBR4 exchange VPNv4/VPNv6 addresses, PE node loopback addresses, and labels using EBGP.

Suppose that ASBR3 is configured so that the routes received from ASBR1 have a higher Local Preference, and that ASBR1 is the primary router and ASBR3 is the back-up router. With this configuration, traffic for AS2 exits AS1 through ASBR1 along the best path through ASBR2. This path through ASBR1 and ASBR2 is the best path that ASBR1 advertises to IBGP peers after assigning a label to the path. The VPNv4/VPNv6 labels and PE loopback addresses from AS2 are also advertised along with the label for the best path. ASBR3 installs the best path as the route for any traffic that it must send to AS2. In doing so, ASBR3 ignores the path through ASBR4 that was learned using EBGP.

You can configure the BGP Best External Path on ASBR1 and ASBR3 to provide a back-up path for the traffic between AS1 and AS2 if the link between ASBR1 and ASBR2 fails.

When you configure BGP Best External Path on ASBR1 and ASBR3, ASBR3 calculates a best external path to AS2, through ASBR4, assigns a label to this path, and installs this path into the Label Forwarding Information Base (LFIB) along with the best path. ASBR3 advertises both the best path and the best external path to ASBR1.

ASBR1 uses the best external path received from ASBR3 to calculate a back-up path and assigns a label to it. ASBR1 installs this best external path in the LFIB along with the best path through ASBR2. ASBR1 advertises the best path and best external path to IBGP peers.

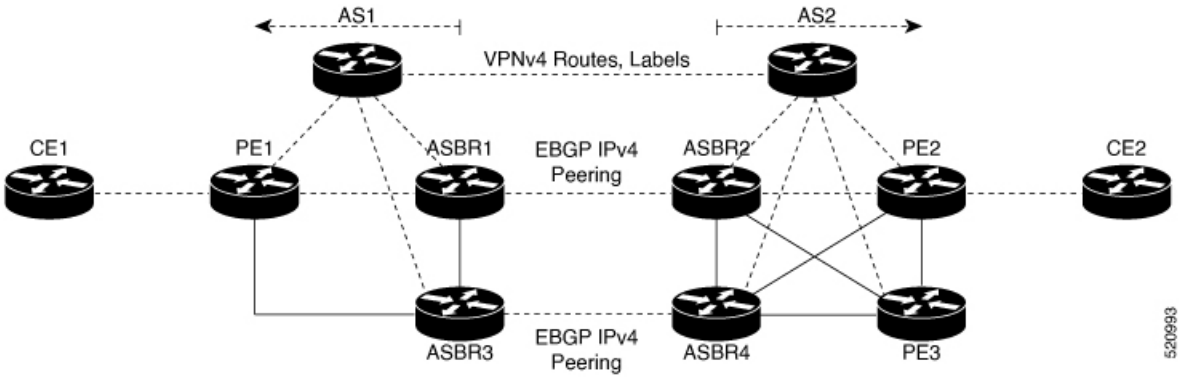
With the best external path installed, suppose PE1 has traffic for PE2 and the ASBR1-ASBR2 link fails, the traffic flows through the best external path along PE1-ASBR1-ASBR3-ASBR4-PE2.

BGP Best External Path with MPLS VPN Inter-AS Option C

With this feature, you can configure the border routers in an MPLS VPN Inter-AS Option C deployment to compute and advertise a best external path based on routes received from EBGP peers. The border routers advertise the best external path so computed to Internal BGP (IBGP) peers through a route reflector. The best external path is advertised in addition to the best path and serves as a back-up path. If a link in the best path fails, traffic flows along the best external path.

Consider the MPLS VPN Inter-AS Option C deployment shown in the following diagram:

Figure 98: MPLS VPN Inter-AS Option C Deployment



The route reflectors, RR1 and RR2, exchange VPNv4 routes and labels using multiprotocol EBGP. The routes preserve VPN labels and next-hop information between ASs. After RR1 reflects the exchanged routes, the next-hop for VPN traffic received by PE1 is either PE2 or PE3.

ASBR1 and ASBR2 exchange IPv4 loopback addresses and labels for PE2 and PE3 using EBGP. Similarly, ASBR3 and ASBR4 IPV4 addresses and labels using EBGP.

Suppose that ASBR3 is configured so that the routes received from ASBR1 have a higher Local Preference, and that ASBR1 is the primary router and ASBR3 is the back-up router. With this configuration, traffic for AS2 exits AS1 through ASBR1 along the best path through ASBR2. This path through ASBR1 and ASBR2 is the best path that ASBR1 advertises to the route reflector RR1 after assigning a label. The PE loopback addresses from AS2 are also advertised to RR1 along with the label for the best path. ASBR3 installs the best path as the route for any traffic that it must send to AS2. In doing so, ASBR3 ignores the path through ASBR4 that was learned using EBGP.

When you configure BGP Best External Path on ASBR1 and ASBR3, ASBR3 calculates a best external path to AS2, through ASBR4, assigns a label to this path, and installs this path into the Label Forwarding Information Base (LFIB) along with the best path. ASBR3 advertises both the best path and the best external path to ASBR1.

ASBR1 uses the best external path received from ASBR3 to calculate a back-up path and assigns a label to it. ASBR1 installs this best external path in the LFIB along with the best path through ASBR2. ASBR1 advertises the best path and the best external path to RR1 for reflection to clients in AS1.

With the best external path installed, suppose PE1 has traffic for PE2 and the ASBR1-ASBR2 link fails, the traffic flows through the best external path along PE1-ASBR1-ASBR3-ASBR4-PE2.

How to Configure BGP Best External

Configuring the BGP Best External Feature

Perform the following task to configure the BGP Best External feature. This task shows how to configure the BGP Best External feature in either an IPv4 or VPNv4 address family. In VPNv4 address family configuration mode, the BGP Best External feature applies to all IPv4 Virtural Routing Forwarding (VRF); you need not

configure it for specific VRFs. If you issue the **bgp advertise-best-external** command in IPv4 VRF address family configuration mode, the BGP Best External feature applies only to that VRF.

Before you begin

- Configure the MPLS VPN and verify that it is working properly before configuring the BGP Best External feature. See the "Configuring MPLS Layer 3 VPNs" section for more information.
- Configure multiprotocol VRFs to allow you to share route-target policies (import and export) between IPv4 and IPv6 or configure separate route-target policies for IPv4 and IPv6 VPNs. For information about configuring multiprotocol VRFs, see the "MPLS VPN--VRF CLI for IPv4 and IPv6 VPNs section".
- Ensure that the customer edge (CE) router is connected to the network by at least two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Do one of the following:
   - **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
   - or
   - **address-family vpnv4** [**unicast**]
   - or
5. **bgp advertise-best-external**
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **activate**
8. **neighbor** *ip-address* **fall-over** [**bfd** | **route-map** *map-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
	• or
	• **address-family vpnv4** [**unicast**]
	• or
	Example:
	`Router(config-router)# address-family ipv4 unicast`
	Example:
	`Router(config-router)# address-family vpnv4`
		Specifies the IPv4 or VPNv4 address family and enters address family configuration mode.
		• The **unicast** keyword specifies the IPv4 or VPNv4 unicast address family.
		• The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.

Step 5	**bgp advertise-best-external** **Example:** `Router(config-router-af)# bgp advertise-best-external`	Calculates and uses an external backup path and installs it into the RIB and Cisco Express Forwarding.
Step 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.
Step 8	**neighbor** *ip-address* **fall-over** [**bfd** | **route-map** *map-name*] **Example:** `Router(config-router-af)# neighbor 192.168.1.1 fall-over bfd`	Configures the BGP peering to use fast session deactivation and enables BFD protocol support for failover. • BGP will remove all routes learned through this peer if the session is deactivated.
Step 9	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits address family configuration mode and returns to privileged EXEC mode.

Verifying the BGP Best External Feature

Perform the following task to verify that the BGP Best External feature is configured correctly.

SUMMARY STEPS

1. **enable**
2. **show vrf detail**
3. **show ip bgp ipv4 mdt all** | **rd vrf**} | **multicast** | **tunnel unicast** or **show ip bgp vpn4 all rd** *route-distinguisher* | **vrf** *vrf-name* **rib-failure** *ip-prefix/length* **longer-prefixes**]] *network-address mask* **longer-prefixes**]] **cidr-only community** *community-list* **dampened-paths filter-list**] [**flap-statistics inconsistent-as neighbors paths line**]] **peer-group quote-regexp regexp**] [**summary labels**
4. **show bgp vpnv4 unicast vrf** *vrf-name ip-address*
5. **show ip route vrf** *vrf-name* **repair-paths** *ip-address*
6. **show ip cef vrf** *vrf-name ip-address* **detail**

DETAILED STEPS

Step 1 enable

Use this command to enable privileged EXEC mode. Enter your password, if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 show vrf detail

Use this command to verify that the BGP Best External feature is enabled. The following **show vrf detail** command output shows that the BGP Best External feature is enabled.

Example:

```
Router# show vrf detail
VRF test1 (VRF Id = 1); default RD 400:1; default VPNID <not set>
  Interfaces:
    Se4/0
Address family ipv4 (Table ID = 1 (0x1)):
  Export VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
    RT:400:1
  Import VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
    RT:400:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix

 Prefix protection with additional path enabled
Address family ipv6 not active.
```

Step 3 show ip bgp ipv4 mdt all | rd vrf} | multicast | tunnel unicast or show ip bgp vpn4 all rd *route-distinguisher* | vrf *vrf-name* rib-failure *ip-prefix/length* longer-prefixes]] *network-address mask* longer-prefixes]] cidr-only

community *community-list* **dampened-paths filter-list**] [**flap-statistics inconsistent-as neighbors paths line**]] **peer-group quote-regexp regexp**] [**summary labels**

Use this command to verify that the best external route is advertised. In the command output, the code b indicates a backup path and the code x designates the best external path.

Example:

```
Router# show ip bgp vpnv4 all
BGP table version is 1104964, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, multipath,
b backup-path, x best-external
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 11:12 (default for vrf blue)
*>i1.0.0.1/32       10.10.3.3                0     200      0 1 ?
* i                 10.10.3.3                0     200      0 1 ?
*                   10.0.0.1                               0 1 ?
*bx                 10.0.0.1                 0             0 1 ?
*                   10.0.0.1                               0 1 ?
```

Step 4 **show bgp vpnv4 unicast vrf** *vrf-name ip-address*

Use this command to verify that the best external route is advertised.

Example:

```
Router# show bgp vpnv4 unicast vrf vpn1 10.10.10.10
BGP routing table entry for 10:10:10.10.10.10/32, version 10
Paths: (2 available, best #1, table vpn1)
  Advertise-best-external
  Advertised to update-groups:
    1          2
  200
    10.6.6.6 (metric 21) from 10.6.6.6 (10.6.6.6)
    Origin incomplete, metric 0, localpref 200, valid, internal, best
    Extended Community: RT:1:1
    mpls labels in/out 23/23
  200
    10.1.2.1 from 10.1.2.1 (10.1.1.1)
    Origin incomplete, metric 0, localpref 100, valid,
external, backup/repair,     advertise-best-external
    Extended Community: RT:1:1 , recursive-via-connected
    mpls labels in/out 23/nolabel
```

Step 5 **show ip route vrf** *vrf-name* **repair-paths** *ip-address*

Use this command to display the repair route.

Example:

```
Router# show ip route vrf vpn1 repair-paths

Routing Table: vpn1
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP
```

```
       + - replicated route, % - next hop override
Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B        10.1.1.0/24 [200/0] via 10.6.6.6, 00:38:33
                     [RPR][200/0] via 10.1.2.1, 00:38:33
B        10.1.1.1/32 [200/0] via 10.6.6.6, 00:38:33
                     [RPR][200/0] via 10.1.2.1, 00:38:33
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.2.0/24 is directly connected, Ethernet0/0
L        10.1.2.2/32 is directly connected, Ethernet0/0
B        10.1.6.0/24 [200/0] via 10.6.6.6, 00:38:33
                      [RPR][200/0] via 10.1.2.1, 00:38:33
```

Step 6 **show ip cef vrf** *vrf-name ip-address* **detail**

Use this command to display the best external route.

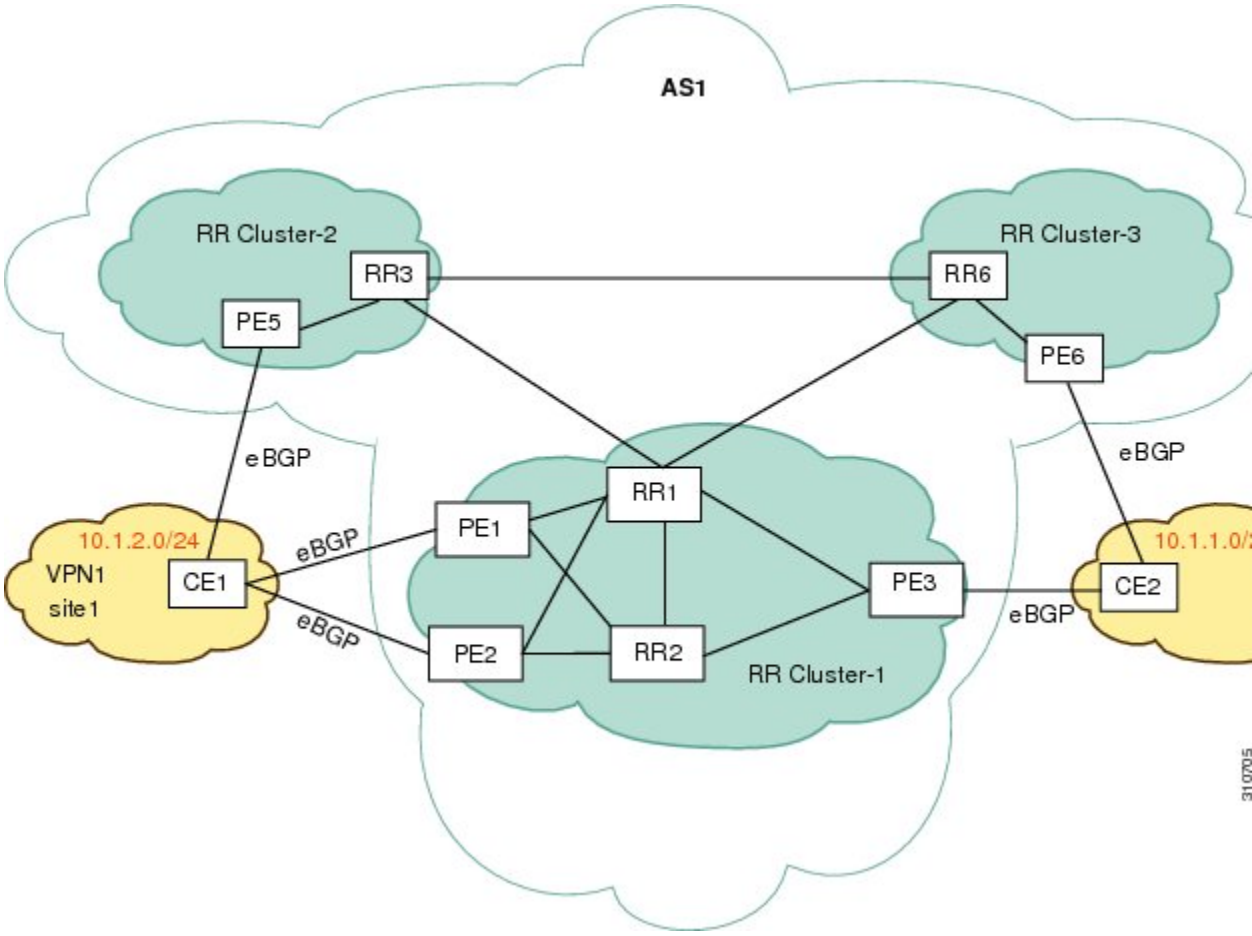
Example:

```
Router# show ip cef vrf test 10.71.8.164 detail

10.71.8.164/30, epoch 0, flags rib defined all labels
 recursive via 10.249.0.102 label 35
   nexthop 10.249.246.101 Ethernet0/0 label 25
 recursive via 10.249.0.104 label 28,
repair
   nexthop 10.249.246.101 Ethernet0/0 label 24
```

Configuring Best External Path on an RR for an Intercluster

Perform the following task to configure a best external path on an RR for an intercluster. The steps in this particular task configure RR1 in the figure below, in the IPv4 address family. The step that configures address family lists the other address families supported.

Figure 99: Scenario for Configuring a BGP Best External Path on a RR for an Intercluster



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **address-family ipv4 unicast**
7. **neighbor** *ip-address* **activate**
8. **neighbor** *ip-address* **activate**
9. **bgp additional-paths select best-external**
10. **bgp additional-paths install**
11. **neighbor** *ip-address* **advertise best-external**
12. **neighbor** *ip-address* **advertise best-external**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 1`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.5.1.1 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • This step is for RR3.
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.5.1.2 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • This step is for RR6.
Step 6	**address-family ipv4 unicast** **Example:** `Router(config-router)# address-family ipv4 unicast`	Specifies the address family and enters address family configuration mode. • Supported address families are ipv4 unicast, vpnv4 unicast, ipv6 unicast, vpnv6 unicast, ipv4+label, and ipv6+label.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 10.5.1.1 activate`	Enables the exchange of information with a BGP neighbor. • This step is for RR3.
Step 8	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 10.5.1.2 activate`	Enables the exchange of information with a BGP neighbor. • This step is for RR6.
Step 9	**bgp additional-paths select best-external** **Example:**	Configures the system to calculate a best external path (external to RR cluster).

	Command or Action	Purpose
	`Router(config-router-af)# bgp additional-paths select best-external`	
Step 10	**bgp additional-paths install** **Example:** `Router(config-router-af)# bgp additional-paths install`	Enables BGP to calculate a backup path for a given address family and to install it into the RIB and CEF. • This step is necessary if the RR is enabled for forwarding (the RR is in the forwarding plane). Otherwise, this step is unnecessary.
Step 11	**neighbor** *ip-address* **advertise best-external** **Example:** `Router(config-router-af)# neighbor 10.5.1.1 advertise best-external`	(Optional) Configures a neighbor to receive the best external path in an advertisement. • This step is for RR3.
Step 12	**neighbor** *ip-address* **advertise best-external** **Example:** `Router(config-router-af)# neighbor 10.5.1.2 advertise best-external`	(Optional) Configures a neighbor to receive the best external path in an advertisement. • This step is for RR6.
Step 13	**end** **Example:** Router(config-router-af)# end	(Optional) Exits address family configuration mode and returns to privileged EXEC mode.

In the scenario shown above, the following paths are selected as best path, backup bath, and best internal path on the three RRs located in the three different clusters:

On RR1:

On RR3:

On RR6:

To Reach Prefix 10/8	Next Hop:
	PE5 (best path, local preference = 200)
	PE3 (backup path, local preference = 150)
	PE3 (best internal path, local preference = 150)

To Reach Prefix 10/8	Next Hop:
	PE5 (best path, local preference = 200)
	PE6 (backup path, local preference = 50)
	PE3 (received as best external path from RR1, local preference = 150)

To Reach Prefix 10/8	Next Hop:
	PE5 (best path, local preference = 200)
	PE6 (backup path, local preference = 50)
	PE3 (received as best external path from RR1, local preference = 150)

Configure BGP Best External Path with MPLS VPN Inter-AS Option B

The configuration instructions in this section are limited to the additional configuration required to enable the computation and advertisement of BGP Best External Path in an MPLS VPN Inter-AS Option B deployment.

To configure BGP Best External Path,

- *Configure the Primary ASBR to Compute and Install a Back-up Path*
- *Configure the Secondary ASBR to Compute, Install, and Advertise Best External Path*

Note An ASBR becomes the Primary ASBR when you configure the routes the ASBR propagates to have a higher local preference on other ASBRs.

Configure the Primary ASBR to Compute and Install a Back-up Path

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router bgp** *autonomous-system-number*

Example:

```
Device(config)# router bgp 1
```

Configures a BGP routing process, and enters router configuration mode for the specified routing process.

Step 4 **address-family vpnv4**

Example:

```
Device(config-router)# address-family vpnv4
```

Specifies the VPNv4 address family and enters address family configuration mode.

Step 5 **bgp additional-paths install**

Example:

```
Device(config-router-af)# bgp additional-paths install
```

Computes and installs the best external path in the LFIB, and advertises the path to IBGP peers.

Step 6 **exit-address-family**

Example:

```
Device(config-router-af)# exit-address-family
```

Exits address family configuration mode.

Step 7 **end**

Example:

```
Device(config-router)# end
```

(Optional) Exits to privileged EXEC mode.

Configure the Secondary ASBR to Compute, Install, and Advertise Best External Path

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router bgp** *autonomous-system-number*

Example:

```
Device(config)# router bgp 1
```

Configures a BGP routing process, and enters router configuration mode for the specified routing process.

Step 4 **address-family vpnv4**

Example:

```
Device(config-router)# address-family vpnv4
```

Specifies the VPNv4 address family and enters address family configuration mode.

Step 5 **bgp advertise-best-external**

Example:

```
Device(config-router-af)# bgp advertise-best-external
```

Computes and installs the best external path in the LFIB, and advertises the path to the primary ASBR.

Step 6 **exit-address-family**

Example:

```
Device(config-router-af)# exit-address-family
```

Exits address family configuration mode.

Step 7 **end**

Example:

```
Device(config-router)# end
```

(Optional) Exits to privileged EXEC mode.

Configure BGP Best External Path with MPLS VPN Inter-AS Option C

The configuration instructions in this section are limited to the additional configuration required to enable the computation and advertisement of BGP Best External Path in an MPLS VPN Inter-AS Option C deployment.

To configure BGP Best External Path,

- *Configure the Primary ASBR to Compute and Install a Back-up Path*
- *Configure the Secondary ASBR to Compute, Install, and Advertise Best External Path*

Note An ASBR becomes the Primary ASBR when you configure the routes the ASBR propagates to have a higher local preference on other ASBRs.

Configure the Primary ASBR to Compute and Install a Back-up Path

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router bgp** *autonomous-system-number*

Example:

```
Device(config)# router bgp 1
```

Configures a BGP routing process, and enters router configuration mode for the specified routing process.

Step 4 **address-family ipv4 unicast**

Example:

```
Device(config-router)# address-family ipv4 unicast
```

Specifies the IPv4 unicast address family and enters address family configuration mode.

Step 5 **bgp additional-paths install**

Example:

```
Device(config-router-af)# bgp additional-paths install
```

Computes and installs the best external path in the LFIB, and advertises the path to IBGP peers.

Step 6 **exit-address-family**

Example:

```
Device(config-router-af)# exit-address-family
```

Exits address family configuration mode.

Step 7 **end**

Example:

```
Device(config-router)# end
```

(Optional) Exits to privileged EXEC mode.

Configure the Secondary ASBR to Compute, Install, and Advertise Best External Path

Perform this procedure to configure BGP Best External Path on the Secondary ASBR in an MPLS VPN Inter-AS Option C deployment.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router bgp** *autonomous-system-number*

Example:

```
Device(config)# router bgp 1
```

Configures a BGP routing process, and enters router configuration mode for the specified routing process.

Step 4 **address-family ipv4 unicast**

Example:

```
Device(config-router)# address-family ipv4 unicast
```

Specifies the IPv4 unicast address family and enters address family configuration mode.

Step 5 **bgp advertise-best-external**

Example:

```
Device(config-router-af)# bgp advertise-best-external
```

Computes and installs the best external path in the LFIB, and advertises the path to the primary ASBR.

Step 6 **exit-address-family**

Example:

```
Device(config-router-af)# exit-address-family
```

Exits address family configuration mode.

Step 7 **end**

Example:

```
Device(config-router)# end
```

(Optional) Exits to privileged EXEC mode.

Verify BGP Best External Path with MPLS VPN Inter-AS Option B or MPLS VPN Inter-AS Option C

Verify VPNv4 routes

To check for BGP external paths among the installed VPNv4 routes, issue the command **show bgp vpnv4 unicast all**.

In the command output, the status code **x best-external** indicates that a path is a best external path, as shown in the following example.

```
Device#show bgp vpnv4 unicast all
BGP table version is 12211, local router ID is 44.44.44.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:200
 *>i  19.1.1.1/32      33.33.33.33              0    600      0 200 2001 i
 *b x                  21.1.1.2                               0 200 2001 i

Device#show bgp vpnv4 unicast all 19.1.1.1/32
BGP routing table entry for 200:200:19.1.1.1/32, version 7682
Paths: (2 available, best #1, no table)
  Advertise-best-external
  Advertised to update-groups:
```

```
    1          2
  Refresh Epoch 1
  200 2001
    33.33.33.33 (via default) from 33.33.33.33 (33.33.33.33)
      Origin IGP, metric 0, localpref 600, valid, internal, best
      Extended Community: RT:100:100 , recursive-via-host
      mpls labels in/out 2036/2036
      rx pathid: 0, tx pathid: 0x0
      Updated on Jun 22 2020 20:54:43 PST
  Refresh Epoch 1
  200 2001
    21.1.1.2 (via default) from 21.1.1.2 (22.22.22.22)
      Origin IGP, localpref 100, valid, external, backup/repair, advertise-best-external
      Extended Community: RT:100:100 , recursive-via-connected
      mpls labels in/out 2036/2035
      rx pathid: 0, tx pathid: 0
      Updated on Jun 22 2020 20:54:43 PST
Device#
```

Verify IPv4 LU routes

To check for BGP external paths among the installed IPv4 Labelled Unicast (LU) routes, issue the command **show bgp ipv4 unicast**.

In the command output, the status code **x best-external** indicates that a path is a best external path, as shown in the following example.

```
Device#show bgp ipv4 unicast
BGP table version is 5007, local router ID is 44.44.44.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *bi  11.11.11.11/32   33.33.33.33             11    600      0 i
 *>                    23.1.1.1                11         32768 i
 *>i  66.66.66.66/32   33.33.33.33             11    600      0 200 i
 *b x                  21.1.1.2                11             0 200 i

Device#show bgp ipv4 unicast 66.66.66.66/32
BGP routing table entry for 66.66.66.66/32, version 7
Paths: (2 available, best #1, table default)
  Advertise-best-external
  Advertised to update-groups:
     1          2          3
  Refresh Epoch 1
  200
    33.33.33.33 from 33.33.33.33 (33.33.33.33)
      Origin IGP, metric 11, localpref 600, valid, internal, best
      mpls labels in/out 19/19
      rx pathid: 0, tx pathid: 0x0
      Updated on Jun 22 2020 20:38:02 PST
  Refresh Epoch 1
  200
    21.1.1.2 from 21.1.1.2 (22.22.22.22)
      Origin IGP, metric 11, localpref 100, valid, external, backup/repair,
advertise-best-external , recursive-via-connected
      mpls labels in/out 19/18
      rx pathid: 0, tx pathid: 0
```

```
      Updated on Jun 22 2020 20:37:58 PST
Device#
```

Verify BGP Best External Path Advertisement

To confirm that an ASBR is advertising a best external path for a VPNv4 unicast route, issue the command s**how bgp vpnv4 unicast all neighbor** *neighbour-ip-address* **advertise-route**.

```
Device#show bgp vpnv4 unicast all neighbors 77.77.77.77 advertised-routes
BGP table version is 16792, local router ID is 44.44.44.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:200
 *b x 8.88.88.88/32    21.1.1.2                               0 200 2001 i
 *b x 88.88.88.88/32   21.1.1.2                               0 200 2001 i

Total number of prefixes 2
Device#
```

To confirm that an ASBR is advertising a best external path for an IPv4 LU route, issue the command **show bgp ipv4 unicast all neighbor** *neighbour-ip-address* **advertise-route**.

Verify the Presence of a Best Internal Path and a Best External Path

To confirm that both a best internal path and a best external path are available for a node, issue the command **show ip cef** *dest-ip-address* **internal**.

```
Device#show ip cef 66.66.66.66 internal
66.66.66.66/32, epoch 0, flags [rlbls], RIB[B], refcnt 6, per-destination sharing
  sources: RIB, RR
  feature space:
    IPRM: 0x00018000
    LFD: 66.66.66.66/32 0 local labels
        contains path extension list
        dflt label switch chain 0x7FBDE68C24D0
          loadinfo 7FBDE68C24D0, per-session, 2 choices, flags 0493, 5 locks
  subblocks:
    1 RR source [non-eos indirection, heavily shared]
      non-eos chain loadinfo 7FBDE1787F60, per-session, flags 0591, 3 locks
  ifnums:
    Ethernet0/1(3): 11.1.1.3
    Ethernet1/3(9): 23.1.1.4
```

Configuration Examples for BGP Best External

Example: Configuring the BGP Best External Feature

The following example shows how to configure the BGP Best External feature in VPNv4 mode:

```
vrf definition test1
 rd 400:1
```

```
 route-target export 100:1
 route-target export 200:1
 route-target export 300:1
 route-target export 400:1
 route-target import 100:1
 route-target import 200:1
 route-target import 300:1
 route-target import 400:1
 address-family ipv4
 exit-address-family
 exit
!
interface Ethernet1/0
 vrf forwarding test1
 ip address 10.0.0.1 255.0.0.0
 exit
!
router bgp 64500
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.5.5.5 remote-as 64500
 neighbor 10.5.5.5 update-source Loopback0
 neighbor 10.6.6.6 remote-as 64500
 neighbor 10.6.6.6 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4

bgp advertise-best-external
  neighbor 10.5.5.5 activate
  neighbor 10.5.5.5 send-community extended
  neighbor 10.6.6.6 activate
  neighbor 10.6.6.6 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf test1
  no synchronization
bgp recursion host
  neighbor 192.168.13.2 remote-as 64511
  neighbor 192.168.13.2 fall-over bfd
  neighbor 192.168.13.2 activate
  neighbor 192.168.13.2 as-override
 exit-address-family
```

Example: Configuring a Best External Path on an RR for an Intercluster

The following example configures RR1 in the figure shown in the “Configuring a Best External Path on an RR for an Intercluster” section. RR1 is configured to calculate, install, and advertise the best external path to its intercluster RR neighbors.

RR1

```
router bgp 1
 neighbor 10.5.1.1 remote-as 1
 neighbor 10.5.1.2 remote-as 1
 address-family ipv4 unicast
 neighbor 10.5.1.1 activate
 neighbor 10.5.1.2 activate
 bgp additional-paths select best-external
 bgp additional-paths install
 neighbor 10.5.1.1 advertise best-external
```

```
neighbor 10.5.1.2 advertise best-external
end
```

Example: Configuring BGP Best External Path with MPLS VPN Inter-AS Option B

In this sample topology, EBGP VPNv4 peering exists between ASBR3 and ASBR5, and similarly between ASBR4 and ASBR2.

ASBR3 is configured as the primary ASBR for ASN 100. Thus, traffic flows from CE9 to CE8 through the path CE9 -> PE1 -> RR7 -> ASBR3 -> ASBR5 -> PERR6 -> CE8. In other words, the primary path is through RR7, ASBR3, and ASBR5.

The secondary ASBR, ASBR4, receives the VPNv4 route 88.88.88.88 (CE8) from ASBR3 (best internal path) and ASBR2 (best external path).

With BGP External Path configured, ASBR4 advertises the best external path to both ASBR3 and RR7. ASBR3 advertises the best external path as a back-up path along with the best path through ASBR3 and ASBR5. If the link between ASBR3 and ASBR5 fails, traffic from CE9 to CE8 flows through ASBR4 -> ASBR2 -> PERR6.

ASBR3 Configuration:

```
router bgp 100
 bgp log-neighbor-changes
 no bgp default route-target filter
 neighbor 12.1.1.5 remote-as 200
 neighbor 12.1.1.5 fall-over bfd
 neighbor 44.44.44.44 remote-as 100
 neighbor 44.44.44.44 update-source Loopback0
 neighbor 44.44.44.44 fall-over
 neighbor 77.77.77.77 remote-as 100
 neighbor 77.77.77.77 update-source Loopback0
 neighbor 77.77.77.77 fall-over
 !
 address-family vpnv4
  bgp additional-paths install
  neighbor 12.1.1.5 activate
  neighbor 12.1.1.5 send-community both
  neighbor 12.1.1.5 route-map rt-rewrite-map in
  neighbor 33.33.33.33 activate
  neighbor 33.33.33.33 send-community both
  neighbor 33.33.33.33 next-hop-self
  neighbor 33.33.33.33 route-map LOCPREF-600 in
  neighbor 77.77.77.77 activate
  neighbor 77.77.77.77 send-community both
  neighbor 77.77.77.77 next-hop-self
 exit-address-family
```

ASBR4 Configuration:

```
router bgp 100
 bgp log-neighbor-changes
 no bgp default route-target filter
 neighbor 21.1.1.2 remote-as 200
 neighbor 21.1.1.2 fall-over bfd
 neighbor 33.33.33.33 remote-as 100
 neighbor 33.33.33.33 update-source Loopback0
 neighbor 33.33.33.33 fall-over
 neighbor 77.77.77.77 remote-as 100
 neighbor 77.77.77.77 update-source Loopback0
 neighbor 77.77.77.77 fall-over
 !
 address-family vpnv4
  bgp advertise-best-external
  neighbor 21.1.1.2 activate
  neighbor 21.1.1.2 send-community both
  neighbor 21.1.1.2 route-map rt-rewrite-map in
  neighbor 33.33.33.33 activate
  neighbor 33.33.33.33 send-community both
  neighbor 33.33.33.33 next-hop-self
  neighbor 33.33.33.33 route-map LOCPREF-600 in
  neighbor 77.77.77.77 activate
  neighbor 77.77.77.77 send-community both
  neighbor 77.77.77.77 next-hop-self
 exit-address-family
```

Verification of Best External Path:

ASBR4 advertises the best external path to RR7. This advertisement can be verified using the **show bgp vpnv4 unicast all neighbor** *neighbour-ip-address* **advertise-route** command.

```
ASBR4#show bgp vpnv4 unicast all neighbors 77.77.77.77 advertised-routes
BGP table version is 16792, local router ID is 44.44.44.44
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 200:200
 *b x 8.88.88.88/32    21.1.1.2                               0 200 2001 i
 *b x 88.88.88.88/32   21.1.1.2                               0 200 2001 i

Total number of prefixes 2
ASBR4#
```

Example: Configuring BGP Best External Path with MPLS VPN Inter-AS Option C

In this sample topology, EBGP IPv4 Labelled Unicast (LU) peering is configured between ASBR3 and ASBR5, and similarly, between ASBR4 and ASBR2. EBGP VPNv4 multihop peering is configured between PERR1 and PERR6.

ASBR3 is configured as the primary ASBR for ASN 100. Thus, traffic flows from CE7 to CE8 through the path CE7 -> PERR1 -> ASBR3 -> ASBR5 -> PERR6 -> CE8. In other words, the primary path is through RR7, ASBR3, and ASBR5.

The secondary ASBR, ASBR4, receives the BGP route 66.66.66.66 (PERR6) from ASBR3 (best internal path) and ASBR2 (best external path).

With BGP External Path configured, ASBR4 advertises the best external path to both ASBR3 and PERR1. ASBR3 advertises the best external path as a back-up path along with the best path through ASBR3 and ASBR5. If the link between ASBR3 and ASBR5 fails, traffic from CE7 to CE8 flows through ASBR4 -> ASBR2 -> PERR6.

ASBR3 Configuration:

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 11.11.11.11 remote-as 100
 neighbor 11.11.11.11 update-source Loopback0
 neighbor 12.1.1.5 remote-as 200
 neighbor 12.1.1.5 fall-over bfd
 neighbor 44.44.44.44 remote-as 100
 neighbor 44.44.44.44 update-source Loopback0
 neighbor 44.44.44.44 fall-over
 !
 address-family ipv4
  bgp redistribute-internal
  bgp additional-paths install
  network 11.11.11.11 mask 255.255.255.255
  neighbor 11.11.11.11 activate
  neighbor 11.11.11.11 next-hop-self all
  neighbor 11.11.11.11 route-map SET-MPLS-LABEL out
  neighbor 11.11.11.11 send-label
  neighbor 12.1.1.5 activate
  neighbor 12.1.1.5 route-map SET-MPLS-LABEL out
  neighbor 12.1.1.5 send-label
  neighbor 44.44.44.44 activate
  neighbor 44.44.44.44 route-reflector-client
  neighbor 44.44.44.44 next-hop-self
  neighbor 44.44.44.44 route-map SET-MPLS-LABEL out
  neighbor 44.44.44.44 send-label
 exit-address-family
```

ASBR4 Configuration:

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 11.11.11.11 remote-as 100
 neighbor 11.11.11.11 update-source Loopback0
 neighbor 11.11.11.11 fall-over
 neighbor 21.1.1.2 remote-as 200
 neighbor 21.1.1.2 fall-over bfd
 neighbor 33.33.33.33 remote-as 100
 neighbor 33.33.33.33 update-source Loopback0
 neighbor 33.33.33.33 fall-over
 !
 address-family ipv4
  bgp advertise-best-external
  network 11.11.11.11 mask 255.255.255.255
  neighbor 11.11.11.11 activate
  neighbor 11.11.11.11 next-hop-self all
  neighbor 11.11.11.11 route-map SET-MPLS-LABEL out
  neighbor 11.11.11.11 send-label
  neighbor 21.1.1.2 activate
  neighbor 21.1.1.2 send-label
  neighbor 33.33.33.33 activate
  neighbor 33.33.33.33 next-hop-self
  neighbor 33.33.33.33 route-map LOCPREF-600 in
  neighbor 33.33.33.33 route-map SET-MPLS-LABEL out
```

```
  neighbor 33.33.33.33 send-label
 exit-address-family
```

Verification of Best External Path:

On PERR1, we can use the command **show ip cef <> internal** to verify that there are two paths to 66.66.66.66 (PERR6).

```
PERR1#show ip cef 66.66.66.66 internal
66.66.66.66/32, epoch 0, flags [rlbls], RIB[B], refcnt 6, per-destination sharing
  sources: RIB, RR
  feature space:
    IPRM: 0x00018000
    LFD: 66.66.66.66/32 0 local labels
        contains path extension list
        dflt label switch chain 0x7FBDE68C24D0
          loadinfo 7FBDE68C24D0, per-session, 2 choices, flags 0493, 5 locks
  subblocks:
    1 RR source [non-eos indirection, heavily shared]
      non-eos chain loadinfo 7FBDE1787F60, per-session, flags 0591, 3 locks
  ifnums:
    Ethernet0/1(3): 11.1.1.3
    Ethernet1/3(9): 23.1.1.4
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Basic MPLS VPNs	"Configuring MPLS Layer 3 VPNs" module in the *MPLS: Layer 3 VPNs Configuration Guide*
Multiprotocol VRFs	"MPLS VPN VRF CLI for IPv4 and IPv6 VPNs" module in the *MPLS: Layer 3 VPNs Configuration Guide*
A failover feature that creates a new path after a link or node failure	MPLS VPN--BGP Local Convergence

Standards

Standard	Title
draft-marques-idr-best-external	*BGP Best External, Advertisement of the best external route to iBGP*

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

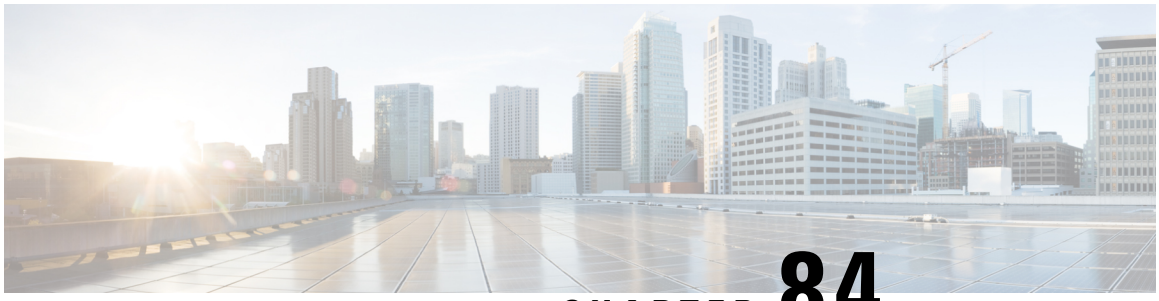
Feature Information for BGP Best External

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 109: Feature Information for BGP Best External

Feature Name	Releases	Feature Information
BGP Best External	Cisco IOS XE Release 3.2S	The BGP Best External feature provides the network with a backup external route to avoid loss of connectivity of the primary external route. This feature advertises the most preferred route among those received from external neighbors as a backup route. In Cisco IOS XE Release 3.2S, this feature was introduced. The following commands were introduced or modified: **bgp advertise-best-external**, **bgp recursion host**, **show ip bgp**, **show ip bgp vpnv4**, **show ip cef**, **show ip cef vrf**, **show ip route**, **show ip route vrf**
BGP Best External Path on an RR for Intercluster	Cisco IOS XE Release 3.4S	The BGP Best External Path on RR for Intercluster feature provides path diversity between RR clusters. The feature provides best external functionality toward non-client iBGP peers, and is also known as "intercluster best external path." The following commands were introduced: **bgp additional-pathsselect**, **neighbor advertise best-external**.
BGP Best External Path with MPLS Inter-AS Options B and C	Cisco IOS XE Amsterdam 17.3.1	In MPLS VPN Inter-AS Option B and MPLS VPN Inter-AS Option C deployments, you can configure border routers to compute and advertise best external paths based on routes received from EBGP peers. The border routers advertise best external paths to Internal BGP (IBGP) peers as back-up paths. If a link in the best path fails, traffic flows along the best external path. This feature is introduced on Cisco ASR 1000 Series Aggregation Services Routers and the Cisco CSR 1000v Cloud Services Router.



CHAPTER 84

BGP PIC Edge for IP and MPLS-VPN

The BGP PIC Edge for IP and MPLS-VPN feature improves BGP convergence after a network failure. This convergence is applicable to both core and edge failures and can be used in both IP and MPLS networks. The BGP PIC Edge for IP and MPLS-VPN feature creates and stores a backup/alternate path in the routing information base (RIB), forwarding information base (FIB), and Cisco Express Forwarding so that when a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover.

Note In this document, the BGP PIC Edge for IP and MPLS-VPN feature is called BGP PIC.

Prerequisites for BGP PIC

- Ensure that the Border Gateway Protocol (BGP) and the IP or Multiprotocol Label Switching (MPLS) network is up and running with the customer site connected to the provider site by more than one path (multihomed).
- Ensure that the backup/alternate path has a unique next hop that is not the same as the next hop of the best path.
- Enable the Bidirectional Forwarding Detection (BFD) protocol to quickly detect link failures of directly connected neighbors.

Restrictions for BGP PIC

The following restrictions apply to the BGP PIC feature:

- With BGP Multipath, the BGP Prefix-Independent Convergence (PIC) feature is already supported.

- In MPLS VPNs, the BGP PIC feature is not supported with MPLS VPN Inter-Autonomous Systems Option B.
- The BGP PIC feature supports prefixes only for IPv4, IPv6, VPNv4, and VPNv6 address families.
- The BGP PIC feature cannot be configured with Multicast or L2VPN Virtual Routing and Forwarding (VRF) address families.
- If the route reflector is only in the control plane, then you do not need BGP PIC, because BGP PIC addresses data plane convergence.
- When two PE devices become each other's backup/alternate path to a CE device, traffic might loop if the CE device fails. Neither device will reach the CE device, and traffic will continue to be forwarded between the PE devices until the time-to-live (TTL) timer expires.
- The BGP PIC feature does not support Nonstop Forwarding with Stateful Switchover (NSF/SSO). However, ISSU is supported if both Route Processors have the BGP PIC feature configured.
- The BGP PIC feature solves the traffic forwarding only for a single network failure at both the edge and the core.
- The BGP PIC feature does not work with the BGP Best External feature. If you try to configure the BGP PIC feature after configuring the BGP Best External feature, you receive an error.

About BGP PIC

In the following sections, we describe the BGP PIC feature in details, how to detect a failure, a scenario and how to configure it.

Benefits

- An extra path for failover allows faster restoration of connectivity when a primary path is invalid or withdrawn.
- Reduction of traffic loss.
- Constant convergence time so that the switching time is the same for all prefixes.

BGP Convergence

Under normal circumstances, BGP can take several seconds to a few minutes to converge after a change in the network. At a high level, BGP goes through the steps of the following process:

1. BGP learns of failures through either Interior Gateway Protocol (IGP) or BFD events or interface events.
2. BGP withdraws the routes from the routing information base (RIB), and the RIB withdraws the routes from the forwarding information base (FIB) and distributed FIB (dFIB). This process clears the data path for the affected prefixes.
3. BGP sends withdrawn messages to its neighbors.
4. BGP calculates the next best path to the affected prefixes.

5. BGP inserts the next best path for affected prefixes into the RIB, and the RIB installs them in the FIB and dFIB.

This process may take from few seconds to a few minutes to complete. It depends on, the latency of the network, the convergence time across the network, and the local load on the devices. The data plane converges only after the control plane converges.

Improve Convergence

The BGP PIC functionality is achieved by an extra functionality in the BGP, RIB, Cisco Express Forwarding, and MPLS.

- BGP Functionality

BGP PIC affects prefixes under IPv4 and VPNv4 address families. For those prefixes, BGP calculates an extra second best path, along with the primary best path. (The second best path is called the backup or alternate path.) BGP installs the best and backup or alternate paths for the affected prefixes into the BGP RIB. The backup or alternate path provides a fast reroute mechanism to counter a singular network failure. BGP also includes the alternate or backup path in its application programming interface (API) to the IP RIB.

- RIB Functionality

For BGP PIC, RIB installs an alternate path per route if one is available. If the RIB selects a BGP route containing a backup or alternate path, it installs the backup or alternate path with the best path. The RIB also includes the alternate path in its API with the FIB.

- Cisco Express Forwarding Functionality

With BGP PIC, Cisco Express Forwarding stores an alternate path per prefix. When the primary path goes down, Cisco Express Forwarding searches for the backup or alternate path in a prefix-independent manner. Cisco Express Forwarding also listens to BFD events to rapidly detect local failures.

- MPLS Functionality

MPLS Forwarding is similar to Cisco Express Forwarding in that it stores alternate paths and switches to an alternate path if the primary path goes down.

When the BGP PIC feature is enabled, BGP calculates a backup or alternate path per prefix and installs it into BGP RIB, IP RIB, and FIB. This improves convergence after a network failure. There are two types of network failures that the BGP PIC feature detects:

- Core node or link failure (internal Border Gateway Protocol [iBGP] node failure): If a PE node or link fails, then the failure is detected through IGP convergence. IGP conveys the failure through the RIB to the FIB.
- Local link or immediate neighbor node failure (external Border Gateway Protocol [eBGP] node or link failure): To detect a local link failure or eBGP single-hop peer node failure in less than a second, you must enable BFD. Cisco Express Forwarding looks for BFD events to detect a failure of an eBGP single-hop peer.

Convergence in the Data Plane

Upon detecting a failure, Cisco Express Forwarding detects the alternate next hop for all prefixes that are affected by the failure. The data plane convergence is achieved in subseconds depending on whether the BGP PIC implementation exists in the software or hardware.

Convergence in the Control Plane

Upon detecting a failure, BGP learns about the failure through IGP convergence or BFD events and sends withdrawn messages for the prefixes, recalculating the best and backup or alternate paths, and advertising the next best path across the network.

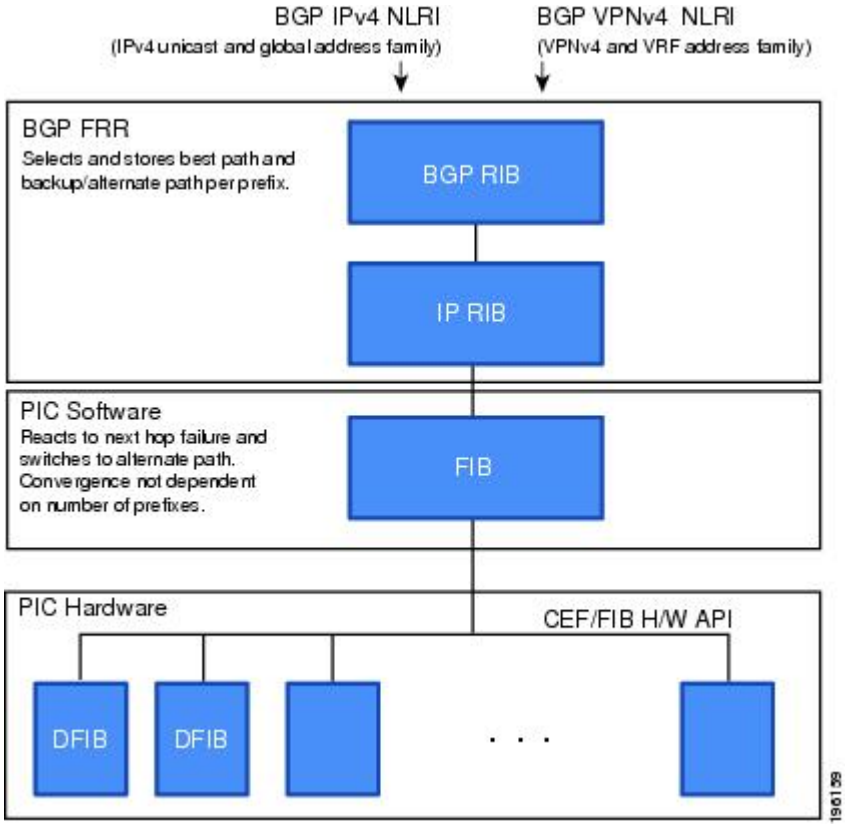
BGP Fast Reroute

BGP Fast Reroute (FRR) provides a best path and a backup or alternate path in BGP, RIB, and Cisco Express Forwarding. BGP FRR provides a fast reroute mechanism into the RIB and Cisco Express Forwarding (CEF) on the backup BGP next hop to reach a destination when the current best path is not available.

BGP FRR precomputes a second best path in BGP and gives it to the RIB and Cisco Express Forwarding as a backup or alternate path, and CEF programs it into line cards.

The BGP PIC feature provides the ability for CEF to quickly switch the traffic to the other egress ports if the current next hop or the link to this next hop goes down.

Figure 100: BGP PIC Edge and BGP FRR



Detect a Failure

IGP detects a failure in the iBGP (remote) peer; it may take a few seconds to detect the failure. Convergence can occur in subseconds or seconds, depending on whether PIC is enabled on the line cards.

If the failure is among the directly connected neighbors (eBGP), and if you use BFD to detect when a neighbor has gone down. Depending on whether PIC is enabled on the line cards, the detection may happen within subseconds and the convergence can occur in subseconds or few seconds.

How BGP PIC Can Achieve Subsecond Convergence

The BGP PIC feature works at the Cisco Express Forwarding level, and Cisco Express Forwarding can be processed in both hardware line cards and in the software.

- For platforms that support Cisco Express Forwarding processing in the line cards, the BGP PIC feature can converge in subseconds.
- For platforms that do not use Cisco Express Forwarding in hardware line cards, Cisco Express Forwarding is achieved in the software. The BGP PIC feature works with the Cisco Express Forwarding through the software and achieves convergence within seconds.

How BGP PIC Improves Upon the Functionality of MPLS VPN BGP Local Convergence

The BGP PIC feature is an enhancement to the "MPLS VPN--BGP Local Convergence" feature, which provides a failover mechanism that recalculates the best path and installs the new path in forwarding after a link failure. The feature maintains the local label for 5 minutes to ensure that the traffic uses the backup/alternate path, thus minimizing traffic loss.

The BGP PIC feature improves the LoC time to under a second by calculating a backup/alternate path in advance. When a link failure occurs, the traffic is sent to the backup/alternate path.

When you configure the BGP PIC feature, it will override the functionality of the "MPLS VPN--BGP Local Convergence" feature. You do not have to remove the **protection local-prefixes** command from the configuration.

Enable BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC allows you to configure, at a time, the BGP PIC feature for all VRFs.

- VPNv4 address family configuration mode protects all VRFs.
- VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
- Router configuration mode protects prefixes in the global routing table.

BGP PIC Scenario

You can configure the BGP PIC functionality to achieve fast convergence.

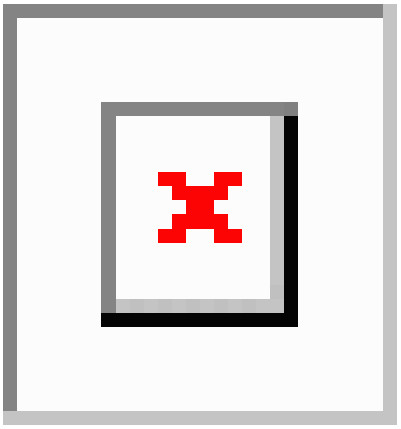
IP PE-CE Link and Node Protection on the CE Side (Dual PEs)

The figure below shows a network that uses the BGP PIC feature. The network includes the following components:

- eBGP sessions exist between the PE and CE devices.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through device CE3.
- CE1 has two paths:
  - PE1 as the primary path.
  - PE2 as the backup/alternate path.

CE1 is configured with the BGP PIC feature. BGP computes PE1 as the best path and PE2 as the backup/alternate path and installs both routes into the RIB and Cisco Express Forwarding plane. When the CE1-PE1 link goes down, Cisco Express Forwarding detects the link failure and points the forwarding object to the backup/alternate path. Traffic is quickly rerouted due to local fast convergence in Cisco Express Forwarding.

Figure 101: Using BGP PIC to Protect the PE-CE Link



IP PE-CE Link and Node Protection on the CE Side (Dual CEs and Dual PE Primary and Backup Nodes)

The figure below shows a network that uses the BGP PIC feature on CE1. The network includes the following components:

- eBGP sessions exist between the PE and CE devices.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through device CE3.
- CE1 has two paths:
  - PE1 as the primary path.
  - PE2 as the backup/alternate path.
- An iBGP session exists between the CE1 and CE2 devices.

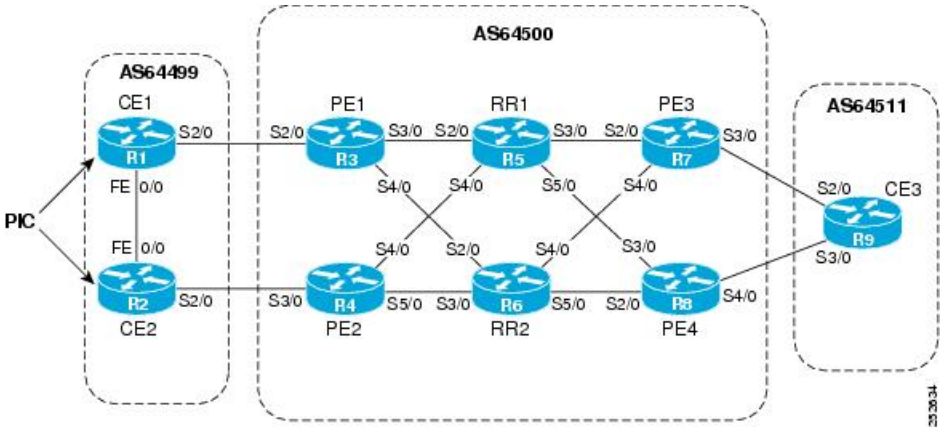
In this example, CE1 and CE2 are configured with the BGP PIC feature. BGP computes PE1 as the best path and PE2 as the backup/alternate path and installs both the routes into the RIB and Cisco Express Forwarding plane.

There should not be any policies set on CE1 and CE2 for the eBGP peers PE1 and PE2. Both CE devices must point to the eBGP route as next hop. On CE1, the next hop to reach CE3 is through PE1, so PE1 is the best path to reach CE3. On CE2, the best path to reach CE3 is PE2. CE2 advertises itself as the next hop to CE1, and CE1 does the same to CE2. As a result, CE1 has two paths for the specific prefix and it usually selects the directly connected eBGP path over the iBGP path according to the best path selection rules. Similarly, CE2 has two paths--an eBGP path through PE2 and an iBGP path through CE1-PE1.

When the CE1-PE1 link goes down, Cisco Express Forwarding detects the link failure and points the forwarding object to the backup/alternate node CE2. Traffic is quickly rerouted due to local fast convergence in Cisco Express Forwarding.

If the CE1-PE1 link or PE1 goes down and BGP PIC is enabled on CE1, BGP recomputes the best path, removing the next hop PE1 from RIB and reinstalling CE2 as the next hop into the RIB and Cisco Express Forwarding. CE1 automatically gets a backup/alternate repair path into Cisco Express Forwarding and the traffic loss during forwarding is now in subseconds, thereby achieving fast convergence.

Figure 102: Using BGP PIC in a Dual CE, Dual PE Network



IP MPLS PE-CE Link Protection for the Primary or Backup Alternate Path

The figure above shows a network that uses the BGP PIC feature on CE1 and CE2. The network includes the following components:

- eBGP sessions exist between the PE and CE devices.
- The PE devices are VPNv4 iBGP peers with reflect devices in the MPLS network.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through device CE3.
- CE3 is dual-homed with PE3 and PE4.
- PE1 has two paths to reach CE3 from the reflect routers:
  - PE3 is the primary path with the next hop as a PE3 address.
  - PE4 is the backup/alternate path with the next hop as a PE4 address.

In this example, all the PE devices can be configured with the BGP PIC feature under IPv4 or VPNv4 address families.

For BGP PIC to work in BGP for PE-CE link protection, set the policies on PE3 and PE4 for prefixes received from CE3 so that one of the PE devices acts as the primary and the other as the backup/alternate. Usually, this is done using local preference and giving better local preference to PE3. In the MPLS cloud, traffic internally flows through PE3 to reach CE3. Thus, PE1 has PE3 as the best path and PE4 as the second path.

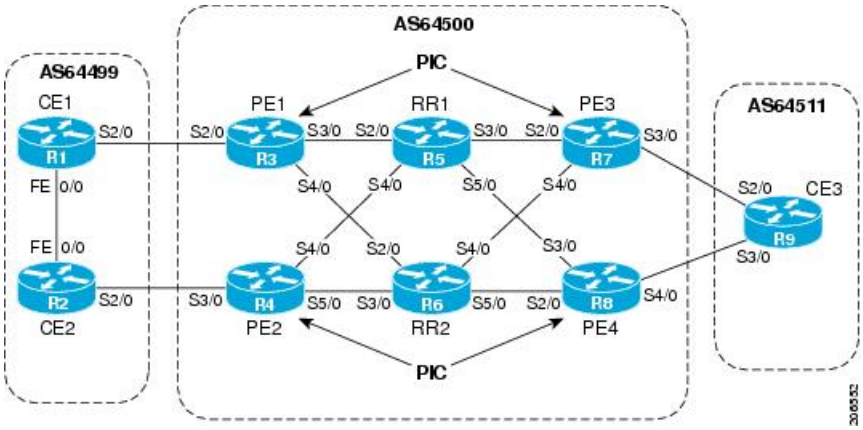
When the PE3-CE3 link goes down, Cisco Express Forwarding detects the link failure, and PE3 recomputes the best path, selects PE4 as the best path, and sends a withdraw message for the PE3 prefix to the reflect routers. Some of the traffic goes through PE3-PE4 until BGP installs PE4 as the best path route into the RIB and Cisco Express Forwarding. PE1 receives the withdraw, recomputes the best path, selects PE4 as the best path, and installs the routes into the RIB and Cisco Express Forwarding plane.

Thus, with BGP PIC enabled on PE3 and PE4, Cisco Express Forwarding detects the link failure and does in-place modification of the forwarding object to the backup/alternate node PE4 that already exists in Cisco Express Forwarding. PE4 knows that the backup/alternate path is locally generated and routes the traffic to the egress port connected to CE3. This way, traffic loss is minimized and fast convergence is achieved.

IP MPLS PE-CE Node Protection for Primary or Backup Alternate Path

The figure below shows a network that uses the BGP PIC feature on all the PE devices in an MPLS network.

Figure 103: Enabling BGP PIC on all PEs devices in the MPLS Network



The network includes the following components:

- eBGP sessions exist between the PE and CE devices.
- The PE devices are VPNv4 iBGP peers with reflect routers in the MPLS network.
- Traffic from CE1 uses PE1 to reach network 192.168.9.0/24 through device CE3.
- CE3 is dual-homed with PE3 and PE4.
- PE1 has two paths to reach CE3 from the reflect routers:
  - PE3 is the primary path with the next hop as a PE3 address.
  - PE4 is the backup/alternate path with the next hop as a PE4 address.

In this example, all the PE devices are configured with the BGP PIC feature under IPv4 and VPNv4 address families.

For BGP PIC to work in BGP for the PE-CE node protection, set the policies on PE3 and PE4 for the prefixes received from CE3 such that one of the PE devices acts as primary and the other as backup/alternate. Usually,

this is done using local preference and giving better local preference to PE3. In the MPLS cloud, traffic internally flows through PE3 to reach CE3. So, PE1 has PE3 as the best path and PE4 as the second path.

When PE3 goes down, PE1 knows about the removal of the host prefix by IGPs in subseconds, recomputes the best path, selects PE4 as the best path, and installs the routes into the RIB and Cisco Express Forwarding plane. Normal BGP convergence will happen while BGP PIC is redirecting the traffic through PE4, and packets are not lost.

Thus, with BGP PIC enabled on PE3, Cisco Express Forwarding detects the node failure on PE3 and points the forwarding object to the backup/alternate node PE4. PE4 knows that the backup/alternate path is locally generated and routes the traffic to the egress port using the backup/alternate path. This way, traffic loss is minimized.

No Local Policies Set on the PE Devices

PE1 and PE2 point to the eBGP CE paths as the next hop with no local policy. Each of the PE devices receives the other's path, and BGP calculates the backup/alternate path and installs it into Cisco Express Forwarding, along with its own eBGP path towards CE as the best path. The limitation of the MPLS PE-CE link and node protection solutions is that you cannot change BGP policies. They should work without the need for a best-external path.

Local Policies Set on the PE Devices

Whenever there is a local policy on the PE devices to select one of the PE devices as the primary path to reach the egress CE, the **bgp advertise-best-external** command is needed on the backup/alternate node PE3 to propagate the external CE routes with a backup/alternate label into the route reflectors and the far-end PE devices.

Cisco Express Forwarding Recursion

Recursion is the ability to find the next longest matching path when the primary path goes down.

If BGP PIC is not installed, and if the next hop to a prefix fails, Cisco Express Forwarding finds the next path to reach the prefix by recursing through the FIB to find the next longest matching path to the prefix. This recurstion mechanism is useful when the next hop is multiple hops away and there is more than one way of reaching the next hop.

However, with the BGP PIC feature, you may want to disable Cisco Express Forwarding recursion for the following reasons:

- Recursion slows down convergence when Cisco Express Forwarding searches all the FIB entries.
- BGP PIC Edge already precomputes an alternate path. It therefore eliminates the need for Cisco Express Forwarding recursion.

When the BGP PIC functionality is enabled, Cisco Express Forwarding recursion is disabled by default for two conditions:

- For next hops learned with a /32 network mask (host routes)
- For next hops that are directly connected.

For all other cases, Cisco Express Forwarding recursion is enabled.

You can issue the **bgp recursion host** command to disable or enable Cisco Express Forwarding recursion for BGP host routes. This provision is part of the BGP PIC functionality.

Note When the BGP PIC feature is enabled, by default, **bgp recursion host** is configured for VPNv4 and VPNv6 address families and disabled for IPv4 and IPv6 address families.

To disable or enable Cisco Express Forwarding recursion for BGP directly connected next hops, run the **disable-connected-check** command.

How to Configure BGP PIC

Configuring BGP PIC

Because many service provider networks contain many VRFs, the BGP PIC feature allows you to configure the BGP PIC feature for all VRFs at once.

- VPNv4 address family configuration mode protects all the VRFs.
- VRF-IPv4 address family configuration mode protects only IPv4 VRFs.
- Router configuration mode protects prefixes in the global routing table.

For a full configuration example that includes configuring multiprotocol VRFs and shows output to verify that the feature is enabled, see the Example: Configuring BGP PIC, on page 1277.

Before you begin

- If you are implementing the BGP PIC feature in an MPLS VPN, ensure that the network is working properly before configuring the BGP PIC feature. See “Configuring MPLS Layer 3 VPNs” for more information.
- If you are implementing the BGP PIC feature in an MPLS VPN, configure multiprotocol VRFs, which allow you to share route-target policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs. For information about configuring multiprotocol VRFs, see “MPLS VPN—VRF CLI for IPv4 and IPv6 VPNs”.
- Ensure that the CE device is connected to the network by at least two paths.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Do one of the following:
   - **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
   - **address-family vpnv4** [**unicast**]
5. **bgp additional-paths install**

6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **activate**
8. **bgp recursion host**
9. **neighbor** *ip-address* **fall-over** [**bfd** |**route-map** *map-name*]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.

Step 4	Do one of the following:
	• **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
	• **address-family vpnv4** [**unicast**]
	Example:
	`Device(config-router)# address-family ipv4 unicast`
	Example:
	`Device(config-router)# address-family vpnv4`
		Specifies the IPv4 or VPNv4 address family and enters address family configuration mode.
		• The **unicast** keyword specifies the IPv4 or VPNv4 unicast address family.
		• The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.

Step 5	**bgp additional-paths install** **Example:** `Device(config-router-af)# bgp additional-paths install`	Calculates a backup/alternate path and installs it into the RIB and Cisco Express Forwarding.
Step 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router. • By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, neighbors must also be activated using the **neighbor activate**

	Command or Action	Purpose
		command in address family configuration mode for the other prefix types.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family with the local router.
Step 8	**bgp recursion host** **Example:** `Device(config-router-af)# bgp recursion host`	(Optional) Enables the recursive-via-host flag for IPv4, VPNv4, and VRF address families. • When the BGP PIC feature is enabled, Cisco Express Forwarding recursion is disabled. Under most circumstances, you do not want to enable recursion when BGP PIC is enabled.
Step 9	**neighbor** *ip-address* **fall-over** [**bfd** |**route-map** *map-name*] **Example:** `Device(config-router-af)# neighbor 192.168.1.1 fall-over bfd`	Enables BFD protocol support to detect when a neighbor has gone away, which can occur within a subsecond.
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Disabling BGP PIC Core

BGP PIC core feature is enabled by default. Use the following configuration to disable the BGP PIC core feature.

Note Use the **cef table output-chain build favor convergence-speed** command in global configuration mode to re-enable the BGP PIC core feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cef table output-chain build favor memory-utilization**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cef table output-chain build favor memory-utilization** **Example:** `Device(config)# cef table output-chain build favor memory-utilization`	Configures memory characteristics for Cisco Express Forwarding table output chain building for the forwarding of packets through the network.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP PIC

Example: Configuring BGP PIC

The following example shows how to configure the BGP PIC feature in VPNv4 address family configuration mode, which enables the feature on all VRFs. In the following example, there are two VRFs defined: blue and green. All the VRFs, including those in VRFs blue and green, are protected by backup/alternate paths.

```
vrf definition test1
 rd 400:1
 route-target export 100:1
 route-target export 200:1
 route-target export 300:1
 route-target export 400:1
 route-target import 100:1
 route-target import 200:1
 route-target import 300:1
 route-target import 400:1
 address-family ipv4
 exit-address-family
 exit
!

 vrf forwarding test1
 ip address 10.0.0.1 255.0.0.0
 exit
```

```
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 redistribute static
 redistribute connected
 neighbor 10.6.6.6 remote-as 3
 neighbor 10.6.6.6 update-source Loopback0
 neighbor 10.7.7.7 remote-as 3
 neighbor 10.7.7.7 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  bgp additional-paths install
  neighbor 10.6.6.6 activate
  neighbor 10.6.6.6 send-community both
  neighbor 10.7.7.7 activate
  neighbor 10.7.7.7 send-community both
 exit-address-family
 !
 address-family ipv4 vrf blue
  import path selection all
  import path limit 10
  no synchronization
  neighbor 10.11.11.11 remote-as 1
  neighbor 10.11.11.11 activate
 exit-address-family
 !
 address-family ipv4 vrf green
  import path selection all
  import path limit 10
  no synchronization
  neighbor 10.13.13.13 remote-as 1
  neighbor 10.13.13.13 activate
 exit-address-family
```

The following **show vrf detail** command output shows that the BGP PIC feature is enabled:

```
Router# show vrf detail
VRF test1 (VRF Id = 1); default RD 400:1; default VPNID <not set>
  Interfaces:
    Se4/0
Address family ipv4 (Table ID = 1 (0x1)):
  Export VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
    RT:400:1
  Import VPN route-target communities
    RT:100:1                RT:200:1                RT:300:1
    RT:400:1
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
  Prefix protection with additional path enabled
Address family ipv6 not active.
```

Example: Displaying Backup Alternate Paths for BGP PIC

The command output in the following example shows that the VRFs in VRF blue have backup/alternate paths:

```
Device# show ip bgp vpnv4 vrf blue 10.0.0.0
```

```
BGP routing table entry for 10:12:12.0.0.0/24, version 88
Paths: (4 available, best #1, table blue)
  Additional-path
  Advertised to update-groups:
     6
  1, imported path from 12:23:12.0.0.0/24
    10.3.3.3 (metric 21) from 10.6.6.6 (10.6.6.6)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
      Extended Community: RT:12:23
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
  1, imported path from 12:23:12.0.0.0/24
    10.13.13.13 (via green) from 10.13.13.13 (10.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, external
      Extended Community: RT:12:23 , recursive-via-connected
  1, imported path from 12:23:12.0.0.0/24
    10.3.3.3 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 200, valid, internal
      Extended Community: RT:12:23
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
  1
    10.11.11.11 from 10.11.11.11 (1.0.0.1)
      Origin incomplete, metric 0, localpref 100, valid, external, backup/repair
      Extended Community: RT:11:12 , recursive-via-connected
```

The command output in the following example shows that the VRFs in VRF green have backup/alternate paths:

```
Device# show ip bgp vpnv4 vrf green 12.0.0.0

BGP routing table entry for 12:23:12.0.0.0/24, version 87
Paths: (4 available, best #4, table green)
  Additional-path
  Advertised to update-groups:
     5
  1, imported path from 11:12:12.0.0.0/24
    10.11.11.11 (via blue) from 10.11.11.11 (1.0.0.1)
      Origin incomplete, metric 0, localpref 100, valid, external
      Extended Community: RT:11:12 , recursive-via-connected
  1
    10.3.3.3 (metric 21) from 10.7.7.7 (10.7.7.7)
      Origin incomplete, metric 0, localpref 200, valid, internal
      Extended Community: RT:12:23
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
  1
    10.13.13.13 from 10.13.13.13 (10.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, external, backup/repair
      Extended Community: RT:12:23 , recursive-via-connected
  1
    10.3.3.3 (metric 21) from 10.6.6.6 (10.6.6.6)
      Origin incomplete, metric 0, localpref 200, valid, internal, best
      Extended Community: RT:12:23
      Originator: 10.3.3.3, Cluster list: 10.0.0.1 , recursive-via-host
      mpls labels in/out nolabel/37
```

The command output in the following example shows the BGP routing table entries for the backup and alternate paths:

```
Device# show ip bgp 10.0.0.0 255.255.0.0

BGP routing table entry for 10.0.0.0/16, version 123
```

```
Paths: (4 available, best #3, table default)
  Additional-path
  Advertised to update-groups:
     2          3
  Local
    10.0.101.4 from 10.0.101.4 (10.3.3.3)
      Origin IGP, localpref 100, weight 500, valid, internal
  Local
    10.0.101.3 from 10.0.101.3 (10.4.4.4)
      Origin IGP, localpref 100, weight 200, valid, internal
  Local
    10.0.101.2 from 10.0.101.2 (10.1.1.1)
      Origin IGP, localpref 100, weight 900, valid, internal, best
  Local
    10.0.101.1 from 10.0.101.1 (10.5.5.5)
      Origin IGP, localpref 100, weight 700, valid, internal, backup/repair
```

The command output in the following example shows the routing information base entries for the backup and alternate paths:

```
Device# show ip route repair-paths 10.0.0.0 255.255.0.0

Routing entry for 10.0.0.0/16
  Known via "bgp 10", distance 200, metric 0, type internal
  Last update from 10.0.101.2 00:00:56 ago
  Routing Descriptor Blocks:
  * 10.0.101.2, from 10.0.101.2, 00:00:56 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: none
    [RPR]10.0.101.1, from 10.0.101.1, 00:00:56 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: none
```

The command output in the following example shows the Cisco Express Forwarding/forwarding information base entries for the backup and alternate paths:

```
Device# show ip cef 10.0.0.0 255.255.0.0 detail

10.0.0.0/16, epoch 0, flags rib only nolabel, rib defined all labels
  recursive via 10.0.101.2
    attached to
  recursive via 10.0.101.1, repair
    attached to
```

Example: Disabling BGP PIC Core

The following example shows how to disable the BGP PIC core in global configuration mode.

```
Device> enable
Device# configure terminal
Device(config)# cef table output-chain build favor memory-utilization
Device(config)# end
```

Additional References

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Basic MPLS VPNs	"Configuring MPLS Layer 3 VPNs" module in the *MPLS: Layer 3 VPNs Configuration Guide*
A failover feature that creates a new path after a link or node failure	"MPLS VPN—BGP Local Convergence" module in the *MPLS: Layer 3 VPNs Configuration Guide*
Configuring multiprotocol VRFs	"MPLS VPN—VRF CLI for IPv4 and IPv6 VPNs" module in the *MPLS: Layer 3 VPNs Configuration Guide*

Related Documents

Standards

Standard	Title
draft-walton-bgp-add-paths-04.txt	*Advertisement of Multiple Paths in BGP*

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

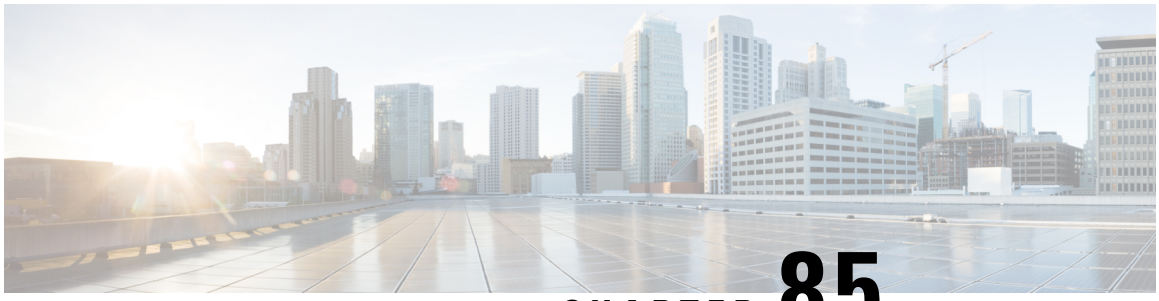
Feature Information for BGP PIC

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 110: Feature Information for BGP PIC

Feature Name	Releases	Feature Information
BGP PIC Edge for IP and MPLS-VPN		The BGP PIC Edge for IP and MPLS-VPN feature improves BGP convergence after a network failure. This convergence is applicable to both core and edge failures and can be used in both IP and MPLS networks. The BGP PIC Edge for IP and MPLS-VPN feature creates and stores a backup/alternate path in the routing information base (RIB), forwarding information base (FIB), and Cisco Express Forwarding so that when a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover. The following commands were introduced or modified: **bgp additional-paths install**, **bgp recursion host**, **show ip bgp**, **show ip cef**, **show ip route**, **show vrf**.



CHAPTER 85

Detecting and Mitigating a BGP Slow Peer

The BGP Slow Peer feature allows a network administrator to detect a BGP slow peer and also to configure a peer as a slow peer statically or to dynamically mark it.

- BGP slow peer detection identifies a BGP peer that is not transmitting update messages within a configured amount of time. It is helpful to know if there is a slow peer, which indicates there is a network issue, such as network congestion or a receiver not processing updates in time, that the network administrator can address.
- BGP slow peer configuration moves or splits the peer from its normal update group to a slow update group, thus allowing the normal update group to function without being slowed down and to converge quickly.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Detecting and Mitigating a BGP Slow Peer

BGP Slow Peer Problem

BGP update generation uses the concept of update groups to optimize performance. An update group is a collection of peers with the identical outbound policy. When generating updates, the group policy is used to format messages that are then transmitted to the members of the group.

In order to maintain fairness in resource utilization, each update group is allocated a quota of formatted messages that it keeps in its cache. Messages are added to the cache when they are formatted by the group, and they are removed when they are transmitted to all the members of the group.

A slow peer is a peer that cannot keep up with the rate at which the Cisco IOS software is generating update messages, and is not keeping up over a prolonged period (in the order of a few minutes). There are several causes of a peer being slow:

- There is packet loss or high traffic on the link to the peer, and the throughput of the BGP TCP connection is very low.
- The peer has a heavy CPU load and cannot service the TCP connection at the required frequency.

When a slow peer is present in an update group, the number of formatted updates pending transmission builds up. When the cache limit is reached, the group does not have any more quotas to format new messages. In order for a new message to be formatted, some of the existing messages must be transmitted by the slow peer and then removed from the cache. The rest of the members of the group that are faster than the slow peer and have completed transmission of the formatted messages will not have anything new to send, even though there may be newly modified BGP networks waiting to be advertised or withdrawn. This effect of blocking formatting of all the peers in a group when one of the peers is slow in consuming updates is the "slow peer" problem.

Temporary Slowness Does Not Constitute a Slow Peer

Events that cause large churn in the BGP table (such as connection resets) can cause a brief spike in the rate of update generation. A peer that temporarily falls behind during such events, but quickly recovers after the event, is not considered a slow peer. In order for a peer for be marked as slow, it must be incapable of keeping up with the average rate of generated updates over a longer period (in the order of a few minutes).

BGP Slow Peer Feature

The BGP Slow Peer feature provides you, the network administrator, with three options:

- You can configure BGP slow peer detection only, which will simply detect a slow peer and provide you with information about it. Such detection is a key feature, especially in a large network of BGP peers, because you can then address the network problem that is causing the slow peer.
- You can configure a dynamic BGP slow peer. When such slow peer protection is configured, slow peer *detection* is enabled by default. The slow peer is moved or "split" from its normal update group to a slow update group, thus allowing the normal update group to function without being slowed down, and to converge more quickly than it would with the slow peer. You have the choice of whether to keep the slow peer in that slow update group until you clear the slow peer (by specifying the **permanent** keyword), or allow the slow peer to dynamically move back to its regular update group as conditions improve. We

recommend that you use the **permanent** keyword and resolve the network issue before you clear the slow peer status.

- You can configure a static BGP slow peer if you already know which peer is slow, perhaps due to a link issue or slow CPU process power. No detection is necessary, and it is more likely that the slow peer will remain there, hence the static configuration.

BGP Slow Peer Detection

You can choose to detect a BGP slow peer, whether or not you also configure the slow peer to be moved to a slow peer update group. Simply detecting a BGP slow peer provides you with useful information about the slow peer without splitting the update group. You should then address the network problem causing the slow peer.

Timestamp on an Update Message

BGP slow peer detection relies on the timestamp on the update messages in an update group. Update messages are timestamped when they are formatted. When BGP slow peer detection is configured, the timestamp of the oldest message in a peers queue is compared to the current time to determine if the peer is lagging more than the configured slow peer time threshold.

For example, if the oldest message in the peers queue was formatted more than 3 minutes ago, but the BGP slow peer detection threshold is configured at 3 minutes, then the peer that formatted that update message is determined to be a slow peer.

The Cisco IOS software generates a syslog event when a slow peer is detected or recovered (when its update group has converged and it has no messages formatted before the threshold time).

Benefit of BGP Slow Peer Detection

Slow peer detection provides you with information about the slow peer, and you can resolve the root cause without moving the peer to a different update group. Therefore, slow peer detection requires just one command that helps you identify something in your network that could be improved.

Benefits of Configuring a Dynamic or Static BGP Slow Peer

When a slow peer is present in an update group, the number of formatted updates pending transmission builds up. New messages cannot be formatted and transmitted until the backlog is reduced. That scenario delays BGP update packets and therefore delays BGP networks from being advertised. The problem can be resolved or prevented by configuring a dynamic slow peer or a static slow peer. Such configuration causes a slow peer to be put into a new, slow peer update group and thus prevents the slow peer from delaying the BGP peers that are not slow.

Static Slow Peer

If you believe that a peer is slow, you can statically configure the peer to be a slow peer. A static slow peer is recommended for a peer that is known to be slow, perhaps having a slow link or low processing power.

Static slow peer configuration causes the Cisco IOS software to create a separate update group for the peer. If you configure two peers belonging to the same update group as slow, these two peers will be moved into

a single slow peer update group because their policy will match. The slow update group will function at the pace of the slowest of the slow peers.

A static slow peer can be configured in either of two ways:

- At the BGP neighbor (address family) level
- Via a peer policy template

You probably want to determine the root cause of the peer being slow, such as network congestion or a receiver not processing updates in time. A static slow peer is not automatically restored to its original update group. You can restore a static slow peer to its original update group by using the **no neighbor slow-peer split-update-group static**command or the **no slow-peer split-update-group static**command.

Dynamic Slow Peer

An alternative to marking a static slow peer is to configure slow peers dynamically, based on the amount of time that the timestamp of the oldest message in a peers queue lags behind the current time. The default threshold is 300 seconds, and is configurable. We recommend that you specify the optional **permanent** keyword, which causes the peer to remain in the slow peer group while you resolve the root cause of the slow peer. You can then use the **clear bgp slow** command to move the peer back to its original group.

If you do not configure the **permanent** keyword, the peer moves back to its original group if and when it regains its non-slow functioning.

When a dynamic slow peer is configured, detection is enabled automatically.

You can configure dynamic slow peers in three ways:

- At the address family view level
- At the neighbor topology (that is, neighbor address-family) level
- Via a peer policy template

How to Detect and Mitigate a BGP Slow Peer

Detecting a Slow Peer

You might want to just detect a slow peer, but not move the slow peer out of its update group. Such detection notifies you by way of a syslog message that a BGP peer is not transmitting update messages within a configurable amount of time. The peer remains in its update group; the update group is not split. The syslog message level is notice level for both detection and recovery.

If you want to dynamically configure a BGP slow peer, see the . YMarking a Peer as a Static Slow Peer, on page 1290ou will notice that that task includes and requires the step of detecting a slow peer.

Detect a slow peer by performing one of the following tasks:

Detecting Dynamic Slow Peers at the Address-Family Level

Perform this task to detect all dynamic slow peers at the address-family level. (If you want to detect *specific* slow peers, detect slow peers at the neighbor level or by using a peer policy template).

The last step is optional; use it if you want to disable slow peer detection for a specific peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4**
6. **bgp slow-peer detection** [**threshold** *seconds*]
7. **neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer detection disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.4.4.4 remote-as 5`	(Optional) Adds an entry to the BGP or multiprotocol BGP neighbor table. • This step is required if you intend to disable dynamic slow peer protection for a specific peer as shown in Step 7 below.
Step 5	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.
Step 6	**bgp slow-peer detection** [**threshold** *seconds*] **Example:** `Router(config-router-af)# bgp slow-peer detection threshold 600`	Configures global slow peer detection and specifies the time in seconds that the timestamp of the oldest update message in a peers queue can be lagging behind the current time before the peer is determined to be a slow peer. • The range of the threshold is from 120 to 3600. As long as the command is configured, the default is 300.

	Command or Action	Purpose
Step 7	**neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer detection disable** **Example:** `Router(config-router-af)# neighbor 10.4.4.4 slow-peer detection disable`	(Optional) Disables slow-peer detection for a specific peer. • Use this command only if you have configured global slow peer detection in Step 5, and now you want to disable slow peer detection for a specific peer or peer group.

Detecting Dynamic Slow Peers at the Neighbor Level

Perform this task to detect dynamic slow peers at a specific neighbor address or belonging to a specific peer group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4**
5. **neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer detection**[**threshold** *seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.
Step 4	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.
Step 5	**neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer detection**[**threshold** *seconds*] **Example:**	(Optional) Specifies the time in seconds that the timestamp of the oldest message in a peers queue can be lagging behind the current time before the peer is determined to be a slow peer.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor 172.60.2.3 slow-peer detection threshold 1200`	• The range of the threshold is 120 seconds to 3600 seconds. As long as the command is configured, the default is 300 seconds.

Detecting Dynamic Slow Peers Using a Peer Policy Template

Perform the following task to detect BGP slow peers using a peer policy template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **slow-peer detection** [**threshold** *seconds*]
6. **exit**
7. **address-family ipv4**
8. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.
Step 4	**template peer-policy** *policy-template-name* **Example:** `Router(config-router)# template peer-policy global`	Enters policy template configuration mode and creates a peer policy template.

	Command or Action	Purpose
Step 5	**slow-peer detection** [**threshold** *seconds*] **Example:** `Router(config-router-ptmp)# slow-peer detection threshold 600`	Specifies the time in seconds that the timestamp of the oldest message in a peers queue can be lagging behind the current time before the peer is determined to be a slow peer. • The range of the threshold is from 120 to 3600. As long as the command is configured, the default is 300.
Step 6	**exit** **Example:** `Router(config-router-ptmp)# exit`	Exits to higher configuration mode.
Step 7	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.
Step 8	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 inherit peer-policy global`	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.

Marking a Peer as a Static Slow Peer

There are two ways to statically configure a slow peer. Perform one of the following tasks in this section to statically configure a slow peer:

Marking a Peer as a Static Slow Peer at the Neighbor Level

Perform this task to configure a static slow peer at a specific neighbor address or belonging to a specific peer group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4**
5. **neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer split-update-group static**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.
Step 4	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.
Step 5	**neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer split-update-group static** **Example:** `Router(config-router-af)# neighbor 172.16.1.1 slow-peer split-update-group static`	Configures the neighbor at the specified address as a slow peer. • Use the **no neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer split-update-group static** command if you want to restore the peer to its original, non-slow update group.

Marking a Peer as a Static Slow Peer Using a Peer Policy Template

Perform this task to configure a static slow peer by using a peer policy template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **slow-peer split-update-group static**
6. **exit**
7. **address-family ipv4**
8. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.
Step 4	**template peer-policy** *policy-template-name* **Example:** `Router(config-router)# template peer-policy global`	Enters policy template configuration mode and creates a peer policy template.
Step 5	**slow-peer split-update-group static** **Example:** `Router(config-router-ptmp)# slow-peer split-update-group static`	Configures the neighbor at the specified address as a slow peer. • Use the **no slow-peer split-update-group static** command if you want to restore the peer to its normal status.
Step 6	**exit** **Example:** `Router(config-router-ptmp)# exit`	Exits to higher configuration mode.
Step 7	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.
Step 8	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 inherit peer-policy global`	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.

Configuring Dynamic Slow Peer Protection

There are three ways to dynamically configure slow peers, also known as slow peer protection. Perform one or more of the tasks in this section to configure dynamic slow peers:

Configuring Dynamic Slow Peers at the Address-Family Level

Configuring dynamic slow peers at the address-family level applies to all peers in the address family specified. (If you want to configure *specific* slow peers, perform this task at the neighbor level or by using a peer policy template.)

The last step is optional; perform it only if you want to disable slow peer protection for a specific peer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number*
5. **address-family ipv4**
6. **bgp slow-peer detection** [**threshold** *seconds*]
7. **bgp slow-peer split-update-group dynamic** [**permanent**]
8. **neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer split-update-group dynamic disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address*[**%**] | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router)# neighbor 10.4.4.4 remote-as 5`	(Optional) Adds an entry to the BGP or multiprotocol BGP neighbor table. • This step is required if you intend to disable dynamic slow peer protection for a specific peer as shown in Step 8 below.
Step 5	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.

	Command or Action	Purpose
Step 6	**bgp slow-peer detection** [**threshold** *seconds*] **Example:** `Router(config-router-af)# bgp slow-peer detection threshold 600`	(Optional) Specifies the time in seconds that the timestamp of the oldest update message in a peers queue can be lagging behind the current time before the peer is determined to be a slow peer. • When a dynamic slow peer is configured, as in the next step, this detection is enabled automatically. • The range of the threshold is from 120 to 3600. The default is 300.
Step 7	**bgp slow-peer split-update-group dynamic** [**permanent**] **Example:** `Router(config-router-af)# bgp slow-peer split-update-group dynamic permanent`	Moves the dynamically detected slow peer to a slow update group. • If a static slow peer update group exists (because of a static slow peer), the dynamic slow peer will be moved to the static slow peer update group. • If no static slow peer update group exists, a new slow peer update group will be created and the peer will be moved to that. • We recommend using the **permanent** keyword. If the **permanent** keyword is used, the peer will not be moved to its original update group automatically. After you determine the root cause of the slowness, such as network congestion, for example, you can use a **clear bgp slow**command to move the peer to its original update group. See the Restoring Dynamic Slow Peers as Normal Peers, on page 1298 to move a dynamically slow peer back to its original update group. • If the **permanent** keyword is not used, the slow peer will be moved back to its regular original update group after it becomes a normal peer (converges).
Step 8	**neighbor** {*neighbor-address*|*peer-group-name*} **slow-peer split-update-group dynamic disable** **Example:** `Router(config-router-af)# neighbor 10.4.4.4 slow-peer split-update-group dynamic disable`	(Optional) Perform this step only if you want to disable dynamic slow peer protection for a specific peer.

Configuring Dynamic Slow Peers at the Neighbor Level

Perform this task to configure a dynamic slow peer at a specific neighbor address or belonging to a specific peer group.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4**
5. **neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer detection** [**threshold** *seconds*]
6. **neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer split-update-group dynamic** [**permanent**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.
Step 4	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.
Step 5	**neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer detection** [**threshold** *seconds*] **Example:** `Router(config-router-af)# neighbor 172.60.2.3 slow-peer detection threshold 1200`	(Optional) Specifies the time in seconds that the timestamp of the oldest update message in a peers queue can be lagging behind the current time before the peer is determined to be a slow peer. • When a dynamic slow peer is configured, as in the next step, this detection is enabled automatically. • The range of the threshold is from 120 to 3600. The default is 300.
Step 6	**neighbor** {*neighbor-address* | *peer-group-name*} **slow-peer split-update-group dynamic** [**permanent**] **Example:** `Router(config-router-af)# neighbor 172.60.2.3 slow-peer split-update-group dynamic permanent`	Moves the dynamically detected slow peer to a slow update group. • If a static slow peer update group exists (because of a static slow peer), the dynamic slow peer will be moved to the static slow peer update group. • If no static slow peer update group exists, a new slow peer update group will be created and the peer will be moved to that.

	Command or Action	Purpose
		• We recommend using the **permanent** keyword. If the **permanent** keyword is used, the peer will not be moved to its original update group automatically. After you resolve the root cause of the slowness, such as network congestion, for example, you can use a **clear bgp slow** command to move the peer to its original update group. See the Restoring Dynamic Slow Peers as Normal Peers, on page 1298 to move a dynamically slow peer back to its original update group. • If the **permanent** keyword is not used, the slow peer will be moved back to its regular original update group after it becomes a normal peer (converges).

Configuring Dynamic Slow Peers Using a Peer Policy Template

Perform this task to configure a BGP slow peer using a peer policy template.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **slow-peer detection** [**threshold** *seconds*]
6. **slow-peer split-update-group dynamic** [**permanent**]
7. **exit**
8. **address-family ipv4**
9. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 5`	Configures the BGP routing process.

	Command or Action	Purpose
Step 4	**template peer-policy** *policy-template-name* **Example:** `Router(config-router)# template peer-policy global`	Enters policy template configuration mode and creates a peer policy template.
Step 5	**slow-peer detection** [**threshold** *seconds*] **Example:** `Router(config-router-ptmp)# slow-peer detection threshold 600`	(Optional) Specifies the time in seconds that the timestamp of the oldest message in a peers queue can be lagging behind the current time before the peer is determined to be a slow peer. • When a dynamic slow peer is configured, as in the next step, this detection is enabled automatically. • The range of the threshold is from 120 to 3600. The default is 300.
Step 6	**slow-peer split-update-group dynamic** [**permanent**] **Example:** `Router(config-router-ptmp)# slow-peer split-update-group dynamic permanent`	Moves the dynamically detected slow peer to a slow update group. • If a static slow peer update group exists (because of a static slow peer), the dynamic slow peer will be moved to the static slow peer update group. • If no static slow peer update group exists, a new slow peer update group will be created and the peer will be moved to that. • We recommend using the **permanent** keyword. If the **permanent** keyword is used, the peer will not be moved to its original update group automatically. After you determine the root cause of the slowness, such as network congestion, for example, you can use a command to move the peer to its original update group. See the Restoring Dynamic Slow Peers as Normal Peers, on page 1298 to move a dynamically slow peer back to its original update group. • If the **permanent** keyword is not used, the slow peer will be moved back to its regular original update group after it becomes a normal peer (converges).
Step 7	**exit** **Example:** `Router(config-router-ptmp)# exit`	Exits to higher configuration mode.
Step 8	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode.

	Command or Action	Purpose
Step 9	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 inherit peer-policy global`	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.

Displaying Output About Dynamic Slow Peers

Use one or more of the **show** commands in this task to display output about dynamically configured BGP slow peers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp** [**ipv4** {**multicast** | **unicast**} | **vpnv4 all** | **vpnv6 unicast all** | **topology**{*****| *routing-topology-instance-name*}] [**update-group**] **summary slow**
3. **show ip bgp** [**ipv4** {**multicast** | **unicast**} | **vpnv4 all** | **vpnv6 unicast all**] **neighbors slow**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp** [**ipv4** {**multicast**	**unicast**}	**vpnv4 all**	**vpnv6 unicast all**	**topology**{*****| *routing-topology-instance-name*}] [**update-group**] **summary slow** **Example:** `Router# show ip bgp summary slow`	Displays information about dynamic BGP slow peers in summary form.
Step 3	**show ip bgp** [**ipv4** {**multicast**	**unicast**}	**vpnv4 all**	**vpnv6 unicast all**] **neighbors slow** **Example:** `Router# show ip bgp neighbors slow`	Displays information about dynamic BGP slow peer neighbors.

Restoring Dynamic Slow Peers as Normal Peers

Once you, the network administrator, resolve the root cause of a slow peer (network congestion, or a receiver not processing updates in time, and so forth), use the **clear**commands in the following task to move the peer back to its original group. Both commands perform the same function.

Note Note that *statically* configured slow peers are not affected by these **clear** commands. To restore a statically configured slow peer to its original update group, use the **no** form of the command shown in one of the tasks in the Marking a Peer as a Static Slow Peer, on page 1290.

SUMMARY STEPS

1. **enable**
2. **clear ip bgp** {[*af*] *****| *neighbor-address* | **peer-group** *group-name*} **slow**
3. **clear bgp** *af* {*****| *neighbor-address* | **peer-group** *group-name*} **slow**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip bgp** {[*af*] *****| *neighbor-address*	**peer-group** *group-name*} **slow** **Example:** `Router# clear ip bgp * slow`	(Optional) Restores neighbor(s) from a slow update peer group to their original update peer group. • *af* is one of the following address families: **ipv4**, **vpnv4**, or **vpnv6**. Moves all peers in the IPv4, VPNv4 or VPNv6 address family back to their original update groups. • ***** moves all peers back to their original update groups.
Step 3	**clear bgp** *af* {*****| *neighbor-address*	**peer-group** *group-name*} **slow** **Example:** `Router# clear bgp ipv4 * slow`	(Optional) Restores neighbor(s) from slow update peer group to their original update peer group. • *af* is one of the following address families: **ipv4**, **vpnv4**, or **vpnv6**. Moves peers in the IPv4, VPNv4 or VPNv6 address family back to their original update groups. • ***** moves all peers in the address family back to their original update groups.

Configuration Examples for Detecting and Mitigating a BGP Slow Peer

Example: Static Slow Peer

The following example marks the neighbor at 192.168.12.10 as a static slow peer.

```
router bgp 5
address-family ipv4
 neighbor 192.168.12.10 slow-peer split-update-group static
```

Example: Static Slow Peer Using Peer Policy Template

The following example configures a static slow peer using a peer policy template named ipv4_ucast_pp2. The neighbor at 10.0.101.4 inherits the policy.

```
router bgp 13
template peer-policy ipv4_ucast_pp2
slow-peer split-update-group static
exit-peer-policy
!
no bgp default route-target filter
no bgp enforce-first-as
bgp log-neighbor-changes
neighbor 10.0.101.4 remote-as 13
address-family ipv4
neighbor 10.0.101.4 inherit peer-policy ipv4_ucast_pp2

RouterA# show ip bgp template peer-policy ipv4_ucast_pp2

Template:ipv4_ucast_pp2, index:2.
Local policies:0x180000000, Inherited polices:0x0
Local disable policies:0x0, Inherited disable policies:0x0
Locally configured policies:
  slow-peer split-update-group static
  Inherited policies:
```

Example: Dynamic Slow Peer at the Neighbor Level

The following example configures a slow peer at the neighbor level. The neighbor at 10.0.101.3 is configured with dynamic slow peer protection at a default threshold of 300 seconds.

```
router bgp 13
no bgp default route-target filter
no bgp enforce-first-as
bgp log-neighbor-changes
 neighbor 10.0.101.3 remote-as 13
address-family ipv4
 neighbor 10.0.101.3 slow-peer split-update-group dynamic
```

Example: Dynamic Slow Peers Using Peer Policy Template

In the following example, Router A uses a peer policy template named ipv4_ucast_pp1 and sets a detection threshold of 120 seconds. The **permanent** keyword causes slow peers to remain in the slow update group until the network administrator uses the **clear ip bgp slow** command to move the peer to its original update group. The neighbor at 10.0.101.2 inherits the peer policy, which means that if that neighbor is determined to be slow, it is moved to a slow update group.

```
router bgp 13
 template peer-policy ipv4_ucast_pp1
 slow-peer detection threshold 120
 slow-peer split-update-group dynamic permanent
 exit-peer-policy
!
no bgp default route-target filter
no bgp enforce-first-as
bgp log-neighbor-changes
 neighbor 10.0.101.2 remote-as 13
!
address-family ipv4
 neighbor 10.0.101.2 activate
 neighbor 10.0.101.2 inherit peer-policy ipv4_ucast_pp1
```

The following output displays the locally configured policies.

```
RouterA# show ip bgp template peer-policy ipv4_ucast_pp1

Template:ipv4_ucast_pp1, index:1.
Local policies:0x300000000, Inherited polices:0x0
Local disable policies:0x0, Inherited disable policies:0x0
Locally configured policies:
  slow-peer detection threshold is 120
  slow-peer split-update-group dynamic permanent
Inherited policies:
```

Example: Dynamic Slow Peers Using Peer Group

The following example configures two peer groups: ipv4_ucast_pg1 and ipv4_ucast_pg2. The neighbor at 10.0.101.1 belongs to ipv4_ucast_pg1, where slow peer detection is configured for 120 seconds. The neighbor at 10.0.101.5 belongs to ipv4_ucast_pg2, where slow peer detection is configured at 140 seconds.

```
router bgp 13
no bgp default route-target filter
no bgp enforce-first-as
bgp log-neighbor-changes
 neighbor ipv4_ucast_pg1 peer-group
 neighbor ipv4_ucast_pg2 peer-group
 neighbor ipv4_ucast_pg1 remote-as 13
 neighbor ipv4_ucast_pg2 remote-as 13
 neighbor 10.0.101.1 peer-group ipv4_ucast_pg1
 neighbor 10.0.101.5 peer-group ipv4_ucast_pg2
address-family ipv4
 neighbor ipv4_ucast_pg1 slow-peer detection threshold 120
 neighbor ipv4_ucast_pg1 slow-peer split-update-group dynamic
 neighbor ipv4_ucast_pg2 slow-peer detection threshold 140
 neighbor ipv4_ucast_pg2 slow-peer split-update-group dynamic
```

The following output displays information about the peer group ipv4_ucast_pg1.

```
RouterA# show ip bgp peer-group ipv4_ucast_pg1

BGP peer-group is ipv4_ucast_pg1,  remote AS 13
  BGP version 4
  Neighbor sessions:
    0 active, is multisession capable
  Default minimum time between advertisement runs is 0 seconds
 For address family: IPv4 Unicast
  BGP neighbor is ipv4_ucast_pg1, peer-group internal, members:
  10.0.101.1
  Index 0
  Slow-peer detection is enabled, threshold value is 120
  Slow-peer split-update-group dynamic is enabled
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
```

The following output displays information about the peer group ipv4_ucast_pg2.

```
RouterA# show ip bgp peer-group ipv4_ucast_pg2

BGP peer-group is ipv4_ucast_pg2,  remote AS 13
  BGP version 4
  Neighbor sessions:
    0 active, is multisession capable
  Default minimum time between advertisement runs is 0 seconds
 For address family: IPv4 Unicast
  BGP neighbor is ipv4_ucast_pg2, peer-group internal, members:
  10.0.101.5
  Index 0
  Slow-peer detection is enabled, threshold value is 140
  Slow-peer split-update-group dynamic is enabled
  Update messages formatted 0, replicated 0
  Number of NLRIs in the update sent: max 0, min 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
MPLS Layer 3 VPN configuration tasks	"Configuring MPLS Layer 3 VPNs" module in the *MPLS: Layer 3 VPNs Configuration Guide*
VRF selection using policy based routing	"MPLS VPN VRF Selection Using Policy-Based Routing" module in the *MPLS: Layer 3 VPNs Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

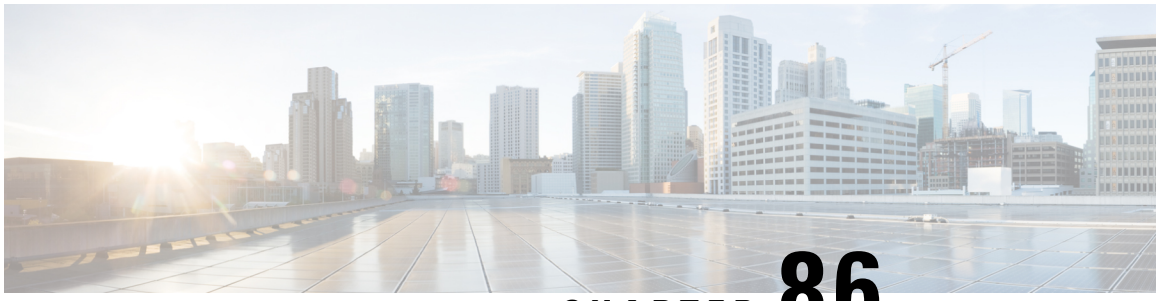
Feature Information for BGP—Support for iBGP Local-AS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 111: Feature Information for BGP—Support for iBGP Local-AS

Feature Name	Releases	Feature Information
BGP—Support for iBGP Local-AS		Prior to the BGP—Support for Local-AS feature, the **neighbor local-as** command was used on a route reflector to customize AS_PATH attributes for routes received from an eBGP neighbor. The **neighbor local-as** command can now be used to enable the sending of the iBGP attributes (LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST) over an iBGP local-AS session. This functionality is useful when merging two autonomous systems, when it is advantageous to keep the iBGP attributes in routes. Prior to the BGP—Support for iBGP Local-AS feature, the RR should not have been configured to change iBGP attributes. With the introduction of this feature, the RR can be configured to change iBGP attributes, providing more flexibility. The following command was introduced: • **neighbor allow-policy** The following commands were modified: • **neighbor local-as** • **show ip bgp vpnv4**



CHAPTER 86

Configuring BGP: RT Constrained Route Distribution

BGP: RT Constrained Route Distribution is a feature that can be used by service providers in Multiprotocol Label Switching (MPLS) Layer 3 VPNs to reduce the number of unnecessary routing updates that route reflectors (RRs) send to Provider Edge (PE) routers. The reduction in routing updates saves resources by allowing RRs, Autonomous System Boundary Routers (ASBRs), and PEs to have fewer routes to carry. Route targets are used to constrain routing updates.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for BGP: RT Constrained Route Distribution

Before you configure BGP: RT Constrained Route Distribution, you should understand how to configure the following:

- Multiprotocol Label Switching (MPLS) VPNs
- Route distinguishers (RDs)

- Route targets (RTs)
- Multiprotocol BGP (MBGP)

Restrictions for BGP: RT Constrained Route Distribution

BGP: RT Constrained Route Distribution constrains all VPN route advertisements.

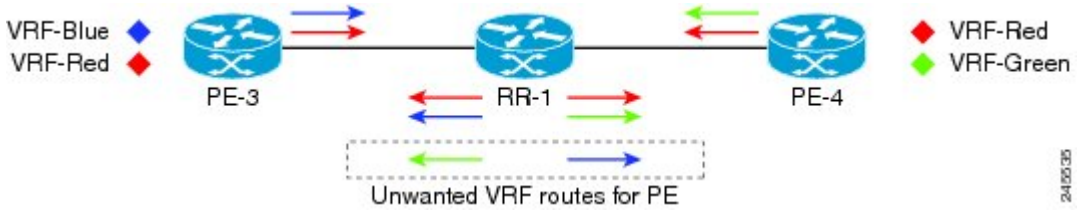
Information About BGP: RT Constrained Route Distribution

Problem That BGP: RT Constrained Route Distribution Solves

Some service providers have a large number of routing updates being sent from RRs to PEs, which can require extensive use of resources. A PE does not need routing updates for VRFs that are not on the PE; therefore, the PE determines that many routing updates it receives are "unwanted." The PE filters out the unwanted updates.

The figure below illustrates a scenario in which unwanted routing updates arrive at two PEs.

Figure 104: Unwanted Routing Updates at PE

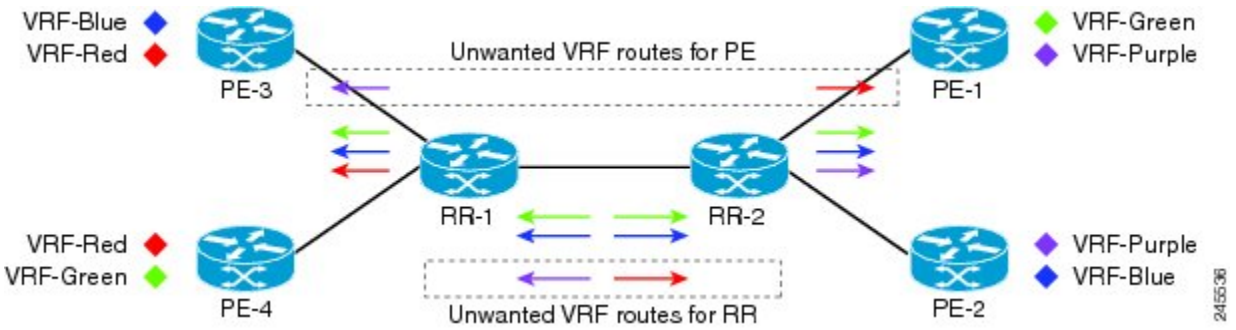


As shown in the figure above, a PE receives unwanted routes in the following manner:

1. PE-3 advertises VRF Blue and VRF Red routes to RR-1. PE-4 advertises VRF Red and VRF Green routes to RR-1.
2. RR-1 has all of the routes for all of the VRFs (Blue, Red, and Green).
3. During a route refresh or VRF provisioning, RR-1 advertises all of the VRF routes to both PE-3 and PE-4.
4. Routes for VRF Green are unwanted at PE-3. Routes for VRF Blue are unwanted at PE-4.

Now consider the scenario where there are two RRs with another set of PEs. There are unwanted routing updates from RRs to PEs and unwanted routing updates between RRs. The figure below illustrates a scenario in which unwanted routes arrive at an RR.

Figure 105: Unwanted Routing Updates at RR



As shown in the figure above, RR-1 and RR-2 receive unwanted routing updates in the following manner:

1. PE-3 and PE-4 advertise VRF Blue, VRF Red, and VRF Green VPN routes to RR-1.
2. RR-1 sends all of its VPN routes to RR-2.
3. VRF Red routes are unwanted on RR-2 because PE-1 and PE-2 do not have VRF Red.
4. Similarly, VRF Purple routes are unwanted on RR-1 because PE-3 and PE-4 do not have VRF Purple.

Hence, a large number of unwanted routes might be advertised among RRs and PEs. The BGP: RT Constrained Route Distribution feature addresses this problem by filtering unwanted routing updates.

Before the BGP: RT Constrained Route Distribution feature, the PE would filter the updates. With this feature, the burden is moved to the RR to filter the updates.

Benefits of BGP: RT Constrained Route Distribution

In MPLS L3VPNs, PE routers use BGP and route target (RT) extended communities to control the distribution of VPN routes to and from VRFs in order to separate the VPNs. PEs and Autonomous System Boundary Routers (ASBRs) commonly receive and then filter out the unwanted VPN routes.

However, receiving and filtering unwanted VPN routes is a waste of resources. The sender generates and transmits a VPN routing update and the receiver filters out the unwanted routes. Preventing the generation of VPN route updates would save resources.

Route Target Constrain (RTC) is a mechanism that prevents the propagation of VPN Network Layer Reachability Information (NLRI) from the RR to a PE that is not interested in the VPN. The feature provides considerable savings in CPU cycles and transient memory usage. RT constraint limits the number of VPN routes and describes VPN membership.

BGP RT-Constrain SAFI

The BGP: RT Constrained Route Distribution feature introduces the BGP RT-Constrain Subsequent Address Family Identifier (SAFI). The command to enter that address family is the **address-family rtfilter unicast** command.

BGP: RT Constrained Route Distribution Operation

In order to filter out the unwanted routes described in the "Problem that BGP RT Constrained Route Distribution Solves" section on page 2, the PEs and RRs must be configured with the BGP: RT Constrained Route Distribution feature.

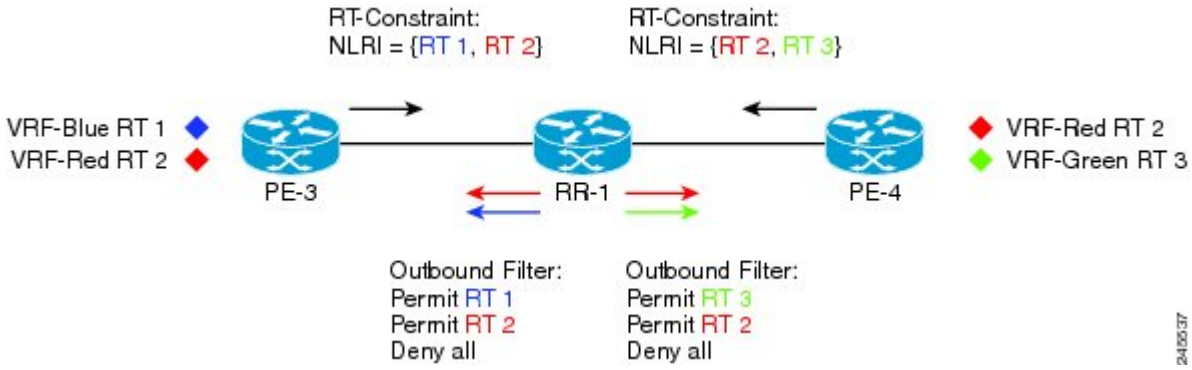
The feature allows the PE to propagate RT membership and use the RT membership to limit the VPN routing information maintained at the PE and RR. The PE uses an MP-BGP UPDATE message to propagate the membership information. The RR restricts advertisement of VPN routes based on the RT membership information it received.

This feature causes two exchanges to happen:

- The PE sends RT Constraint (RTC) Network Layer Reachability Information (NLRI) to the RR.
- The RR installs an outbound route filter.

The figure below illustrates the exchange of the RTC NLRI and the outbound route filter.

Figure 106: Exchange of RTC NLRI and Filter Between PE and RR



As shown in the figure above, the following exchange occurs between the PE and the RR:

1. PE-3 sends RTC NLRI (RT 1, RT 2) to RR-1.
2. PE-4 sends RTC NLRI (RT 2, RT 3) to RR-1.
3. RR-1 translates the NLRI into an outbound route filter and installs this filter (Permit RT 1, RT 2) for PE-3.
4. RR-1 translates the NLRI into an outbound route filter and installs this filter (Permit RT 2, RT 3) for PE-4.

RT Constraint NLRI Prefix

The format of the RT Constraint NLRI is a prefix that is always 12 bytes long, consisting of the following:

- 4-byte origin autonomous system
- 8-byte RT extended community value

The following are examples of RT Constraint prefixes:

- 65000:2:100:1
  - Origin autonomous system number is 65000

- BGP Extended Community Type Code is 2
- Route target is 100:1

- 65001:256:192.0.0.1:100
  - Origin ASN is 65001
  - BGP Extended Community Type Code is 256
  - Route target is 192.0.0.1:100

- 1.10:512:1.10:2
  - Origin ASN is 4-byte, unique 1.10
  - BGP Extended Community Type Code is 512
  - Route target is 1.10:2

To determine what the BGP Extended Community Type Code means, refer to RFC 4360, *BGP Extended Communities Attribute*. In the first example shown, a 2 translates in hexadecimal to 0x002. In RFC 4360, 0x002 indicates that the value that follows the type code will be a two-octet AS specific route target.

RT Constrained Route Distribution Process

This section shows the RT Constrained Route Distribution process. In this example has two CE routers in AS 100 that are connected to PE1. PE1 communicates with PE2, which is also connected to CE routers. Between the two PEs is a route reflector (RR). PE1 and PE2 belong to AS 65000.

The general process for the feature is as follows:

1. The user configures PE1 to activate its BGP peers under the **address-family rtfilter unicast** command.
2. The user configures PE1 in AS 65000 with **route-target import 100:1**, for example.
3. PE1 translates that command to an RT prefix of 65000:2:100:1. The 65000 is the service provider's AS number; the 2 is the BGP Extended Communities Type Code; and the 100:1 is the CE's RT (AS number and another number).
4. PE1 advertises the RT Constrain (RTC) prefix of 65000:2:100:1 to its iBGP peer RR.
5. The RR installs RTC 65000:2:100:1 into the RTC RIB. Each VRF has its own RIB.
6. The RR also installs RTC 65000:2:100:1 into its outbound filter for the neighbor PE1.
7. A filter in the RR either permits or denies the RT. (The AS number is ignored because iBGP is operating in a single AS and does not need to track the AS number.)
8. The RR looks in its outbound filter and sees that it permits outbound VPN packets for RT 100:1 to PE1. So, the RR sends VPN update packet only with RT 100:1 to PE1 and denies VPN updates with any other RT.

Default RT Filter

The default RT filter has a value of zero and length of zero. The default RT filter is used:

- By a peer to indicate that the peer wants all of the VPN routes sent to it, regardless of the RT value.
- By the RR to request that the PE advertise all of its VPN routes to the RR.

The default RT filter is created by configuring the **neighbor default-originate** command under the **address-family rtfilter unicast** command. On the RR it comes as default along with the configuration of route-reflector-client under the address-family rtfilter.

How to Configure RT Constrained Route Distribution

Configuring Multiprotocol BGP on Provider Edge (PE) Routers and Route Reflectors

Perform this task to configure multiprotocol BGP (MP-BGP) connectivity on the PE routers and route reflectors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family vpnv4** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
8. **neighbor** {*ip-address* | *peer-group-name*} **activate**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.

	Command or Action	Purpose
Step 4	**no bgp default ipv4-unicast** **Example:** `Device(config-router)# no bgp default ipv4-unicast`	(Optional) Disables the IPv4 unicast address family on all neighbors. • Use the **no** form of the **bgp default ipv4-unicast** command if you are using this neighbor for MPLS routes only.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor pp.0.0.1 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Device(config-router-af)# neighbor pp.0.0.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor pp.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 9	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp** *ip-address* **events** command, where *ip-address* is the IP address of the neighbor.

Connecting the MPLS VPN Customers

To connect the MPLS VPN customers to the VPN, perform the following tasks:

Defining VRFs on PE Routers to Enable Customer Connectivity

To define virtual routing and forwarding (VRF) instances, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates routing and forwarding tables. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats: • 16-bit AS number: your 32-bit number, for example, 101:3

	Command or Action	Purpose
		• 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the RT extended community attributes to the VRF's list of import, export, or both (import and export) RT extended communities.
Step 6	**import map** *route-map* **Example:** `Device(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	(Optional) Exits to global configuration mode.

Configuring VRF Interfaces on PE Routers for Each VPN Customer

To associate a VRF with an interface or subinterface on the PE routers, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 5/0`	Specifies the interface to configure and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Configuring BGP as the Routing Protocol Between the PE and CE Routers

To configure PE-to-CE routing sessions using BGP, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **exit-address-family**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor pp.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor pp.0.0.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Device(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Configuring RT Constraint on the PE

Perform this task on the PE to configure BGP: RT Constrained Route Distribution with the specified neighbor, and optionally verify that route target (RT) filtering is occurring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family rtfilter unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **activate**
6. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
7. **end**
8. **show ip bgp rtfilter all**
9. **show ip bgp rtfilter all summary**
10. **show ip bgp vpnv4 all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode.

	Command or Action	Purpose
Step 4	**address-family rtfilter unicast** **Example:** `Device(config-router)# address-family rtfilter unicast`	Specifies the RT filter address family type and enters address family configuration mode.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of automated RT filter information with the specified BGP neighbor.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Device(config-router-af)# neighbor pp.0.0.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp rtfilter all** **Example:** `Device# show ip bgp rtfilter all`	(Optional) Displays all BGP RT filter information.
Step 9	**show ip bgp rtfilter all summary** **Example:** `Device# show ip bgp rtfilter all summary`	(Optional) Displays summary BGP RT filter information.
Step 10	**show ip bgp vpnv4 all** **Example:** `Device# show ip bgp vpnv4 all`	(Optional) Displays summary BGP VPNv4 information.

Configuring RT Constraint on the RR

Perform this task on the RR to configure BGP: RT Constrained Route Distribution with the specified neighbor, and optionally verify that route target (RT) filtering is occurring.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family rtfilter unicast**
5. **neighbor** {*ip-address* | *peer-group-name*} **activate**
6. **neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client**
7. **neighbor** {*ip-address* | *peer-group-name*} **send-community extended**
8. **end**
9. **show ip bgp rtfilter all**
10. **show ip bgp rtfilter all summary**
11. **show ip bgp vpnv4 all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode.
Step 4	**address-family rtfilter unicast** **Example:** `Device(config-router)# address-family rtfilter unicast`	Specifies the RT filter address family type and enters address family configuration mode.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 activate`	Enables RT Constraint with the specified BGP neighbor.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client** **Example:**	Enables route-reflector-client funtionality under RT Constraint with the specified BGP neighbor. • Note that the route-reflector-client under RT Constraint address-family comes with a default

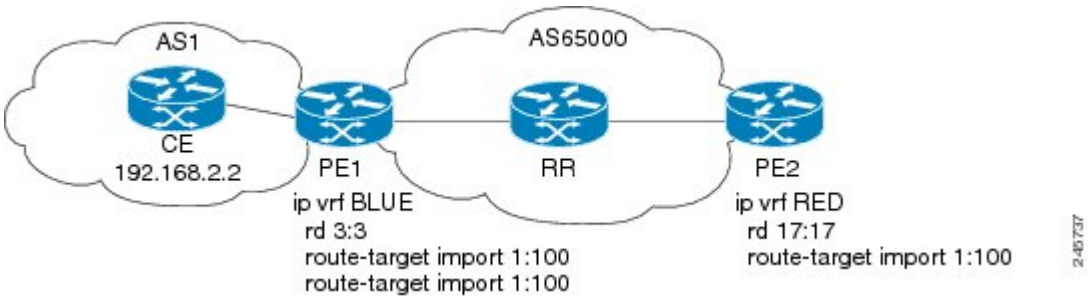
	Command or Action	Purpose
	`Device(config-router-af)# neighbor 10.0.0.2 route-reflector-client`	"neighbor 10.0.0.2 default-originate" functionality that automatically gets added to the BGP configuration. The reason to have this is to have the route-reflector get all the VPN prefixes from its peer.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 9	**show ip bgp rtfilter all** **Example:** `Device# show ip bgp rtfilter all`	(Optional) Displays all BGP RT filter information.
Step 10	**show ip bgp rtfilter all summary** **Example:** `Device# show ip bgp rtfilter all summary`	(Optional) Displays summary BGP RT filter information.
Step 11	**show ip bgp vpnv4 all** **Example:** `Device# show ip bgp vpnv4 all`	(Optional) Displays summary BGP VPNv4 information.

Configuration Examples for BGP: RT Constrained Route Distribution

Example: BGP RT Constrained Route Distribution Between a PE and RR

The following example provides the configurations of the routers in the figure below. PE1 and PE2 are each connected to the RR and belong to AS 65000.

Figure 107: BGP: RT Constrained Route Distribution Between a PE and RR



PE1 Configuration

```
ip vrf BLUE
 rd 3:3
 route-target export 1:100
 route-target import 1:100
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.2.2 remote-as 65000
 neighbor 192.168.2.2 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 send-community extended
 exit-address-family
 !
 address-family rtfilter unicast
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf BLUE
  redistribute static
 exit-address-family
 !
ip route vrf BLUE 51.51.51.51 255.255.255.255 Null0
!
```

RR Configuration

```
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.6.6 remote-as 65000
 neighbor 192.168.6.6 update-source Loopback0
 neighbor 192.168.7.7 remote-as 65000
 neighbor 192.168.7.7 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.6.6 activate
  neighbor 192.168.6.6 send-community extended
  neighbor 192.168.6.6 route-reflector-client
  neighbor 192.168.7.7 activate
  neighbor 192.168.7.7 send-community extended
  neighbor 192.168.7.7 route-reflector-client
```

```
 exit-address-family
 !
 address-family rtfilter unicast
  neighbor 192.168.6.6 activate
  neighbor 192.168.6.6 send-community extended
  neighbor 192.168.6.6 route-reflector-client
  neighbor 192.168.6.6 default-originate
  neighbor 192.168.7.7 activate
  neighbor 192.168.7.7 send-community extended
  neighbor 192.168.7.7 route-reflector-client
  neighbor 192.168.7.7 default-originate
 exit-address-family
!
```

PE2 Configuration

```
!
ip vrf RED
 rd 17:17
 route-target export 150:15
 route-target import 150:1
 route-target import 1:100
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.2.2 remote-as 65000
 neighbor 192.168.2.2 update-source Loopback0
 neighbor 192.168.2.2 weight 333
 no auto-summary
 !
 address-family vpnv4
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 send-community extended
exit-address-family
 !
 address-family rtfilter unicast
  neighbor 192.168.2.2 activate
  neighbor 192.168.2.2 send-community extended
  exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Cisco IOS IP Routing: BGP Command Reference
BGP overview	“Cisco BGP Overview” module
Configuring basic BGP tasks	“Configuring a Basic BGP Network” module

Related Topic	Document Title
BGP fundamentals and description	*Large-Scale IP Network Solutions*, Khalid Raza and Mark Turner, Cisco Press, 2000
Implementing and controlling BGP in scalable networks	*Building Scalable Cisco Networks*, Catherine Paquet and Diane Teare, Cisco Press, 2001
Interdomain routing basics	*Internet Routing Architectures*, Bassam Halabi, Cisco Press, 1997

Standards

Standard	Title
MDT SAFI	MDT SAFI

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1773	*Experience with the BGP Protocol*
RFC 1774	*BGP-4 Protocol Analysis*
RFC 1930	*Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)*
RFC 2519	*A Framework for Inter-Domain Route Aggregation*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 4271	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4684	*Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*
RFC 4893	*BGP Support for Four-Octet AS Number Space*

RFC	Title
RFC 5291	*Outbound Route Filtering Capability for BGP-4*
RFC 5396	*Textual Representation of Autonomous system (AS) Numbers*
RFC 5398	*Autonomous System (AS) Number Reservation for Documentation Use*
RFC 8212	Default External BGP (EBGP) Route Propagation Behavior without Policies

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

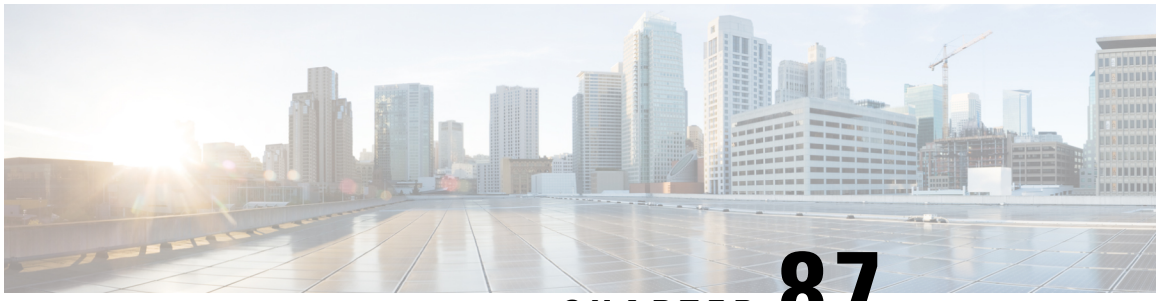
Feature Information for BGP RT Constrained Route Distribution

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 112: Feature Information for BGP: RT Constrained Route Distribution

Feature Name	Releases	Feature Information
BGP: RT Constrained Route Distribution	Cisco IOS XE Release 3.2S	BGP: Route Target (RT) Constrained Route Distribution is a feature that service providers can use in MPLS L3VPNs to reduce the number of unnecessary routes that RRs send to PEs, and thereby save resources. The following commands were introduced: • **address-family rtfilter unicast** • **show ip bgp rtfilter**



CHAPTER 87

Configuring a BGP Route Server

BGP route server is a feature designed for internet exchange (IX) operators that provides an alternative to full eBGP mesh peering among the service providers who have a presence at the IX. The route server provides eBGP route reflection with customized policy support for each service provider. That is, a route server context can override the normal BGP best path for a prefix with a different path based on a policy, or suppress all paths for a prefix and not advertise the prefix. The BGP route server provides reduced configuration complexity and reduced CPU and memory requirements on each border router. The route server also reduces overhead expense incurred by individualized peering agreements.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About BGP Route Server

The Problem That a BGP Route Server Solves

In order to understand the problem that a BGP route server solves, it is helpful to understand the following information about service provider (SP) peering and the eBGP mesh that results from public peering.

Private vs. Public Peering of Service Providers

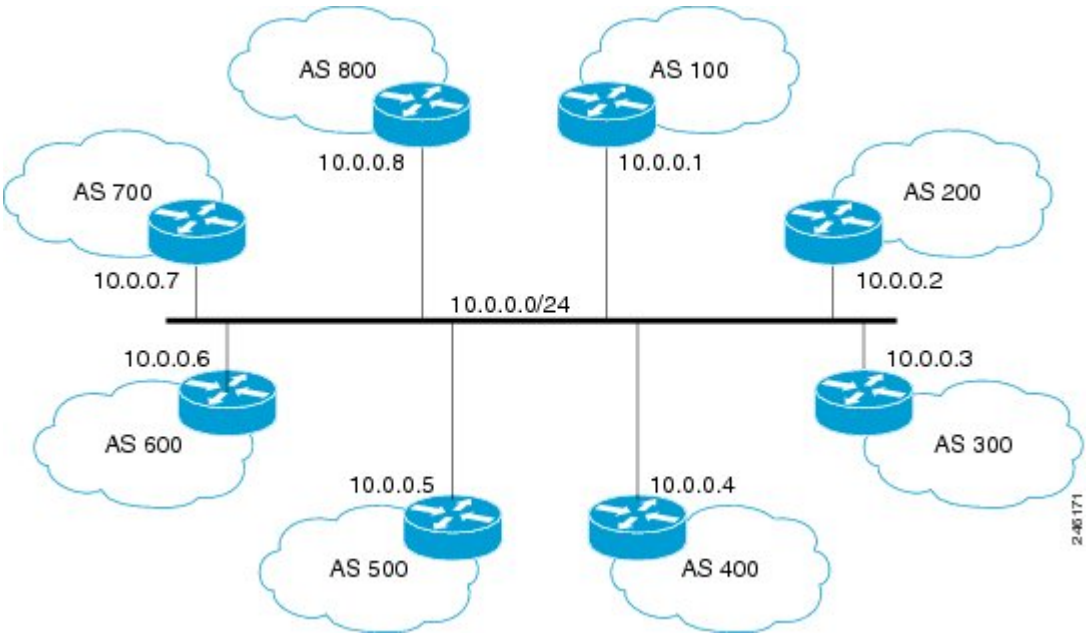
Peering is the connecting of two service providers (SPs) for the purpose of exchanging network traffic between them. Peerings are either private or public.

- In a private peering, two SPs that want to connect decide on a physical site where their networks can be connected and negotiate a contract that covers the details of the connection arrangement. The two parties provide all of the physical space, network equipment, and services (such as electricity and cooling) required to operate the peering connections.
- A public internet exchange (IX), also called a network access point (NAP), is a physical location operated to facilitate the interconnection of multiple SP networks using a shared infrastructure. The IX provides the physical necessities such as rack space for networking devices, electricity, cooling, and a common switching infrastructure required for SPs to directly connect their networks. Unlike private peering, which is typically one-to-one, the IX allows an SP that has a presence at the exchange to connect to multiple peers at a single physical location. The IX provides an alternative to private peering for smaller SPs who do not have the resources required to maintain numerous private peering connections.

Public Peering of SPs within an IX Using BGP

Within the IX, each SP maintains a BGP border router connected to the common switching infrastructure or subnet, as shown in the figure below. In this example, eight different SPs with AS numbers 100 through 800 are connected to the 10.0.0.0/24 subnet through their BGP border routers addressed 10.0.0.1 through 10.0.0.8.

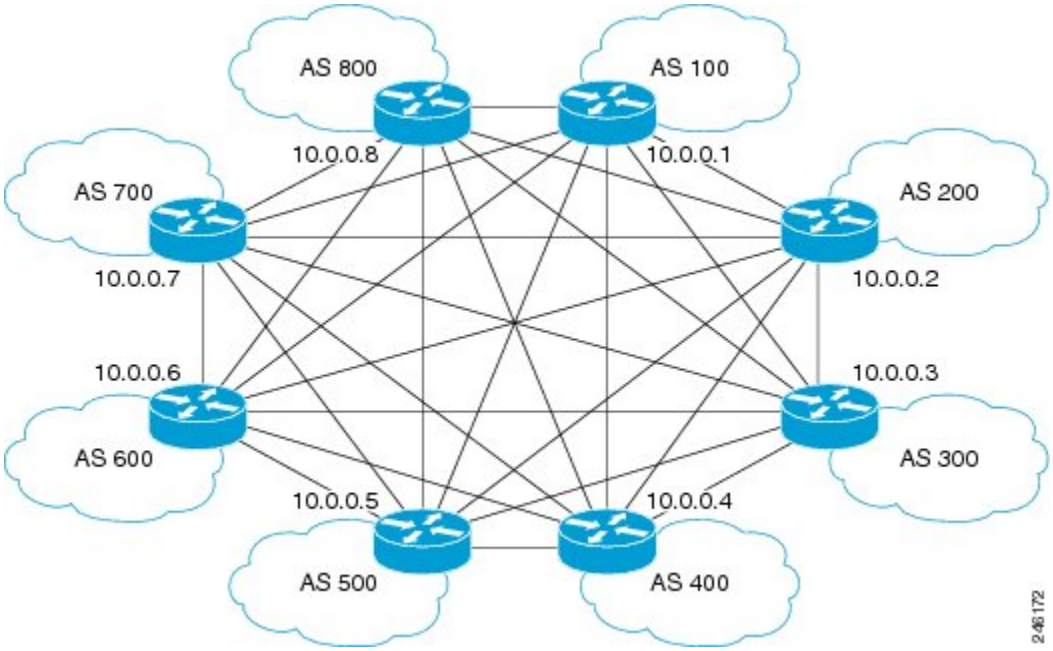
Figure 108: IX Shared Switching Infrastructure



Although each SP's border router is attached to the shared subnet, BGP sessions between each of the SPs must still be configured and maintained individually, for every other SP with which a given SP wants to establish a peering relationship.

Assuming that each SP wants to connect to every other SP, the resulting full mesh of BGP sessions established is shown in the figure below.

Figure 109: IX eBGP Full Mesh



Just as the required iBGP full mesh in an autonomous system presents a scaling and administrative challenge within an SP network, the eBGP full mesh required for peering at an IX presents a challenge for eBGP, for these reasons:

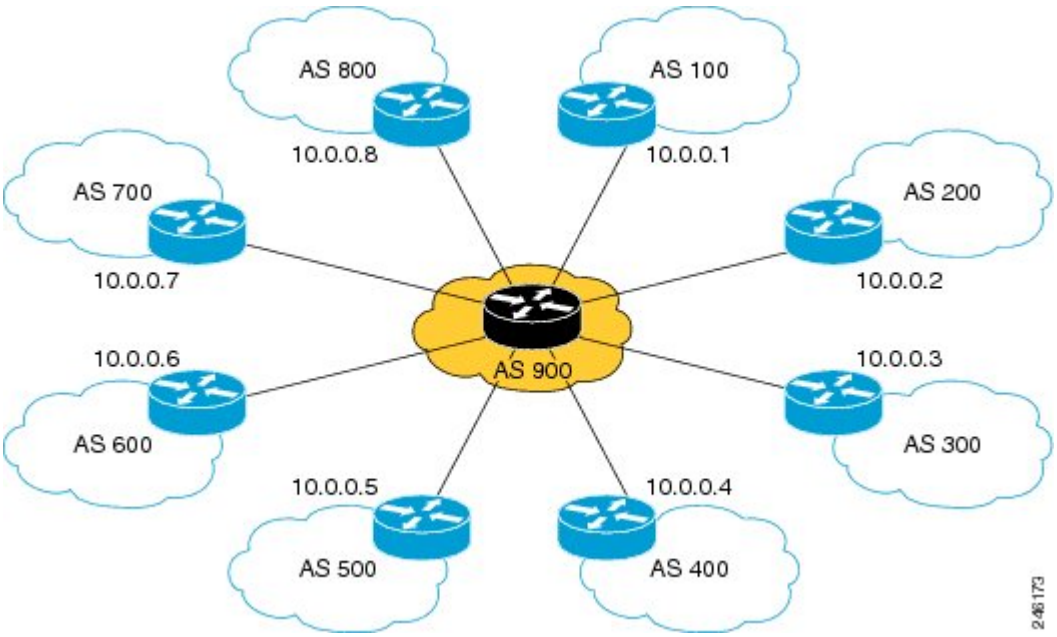
- The full mesh of direct peering sessions requires a BGP session to be configured and maintained for each connection.
- There is additional operational overhead from contracts that would need to be negotiated with each SP peer connecting to a given provider at the IX.

Because larger global SPs might have a presence at dozens or hundreds of internet exchanges worldwide, and dozens or hundreds of potential peers at each IX, it would be a huge operational expense to connect to all of the small providers. Consequently, the state of peering prior to the BGP Route Server feature is that a large global SP connects to only a subset of other large providers to limit the management and operational overhead. A more scalable alternative to direct peering would allow large global SPs to connect to more small providers.

BGP Route Server Simplifies SP Interconnections

A BGP route server simplifies interconnection of SPs at an IX, as shown in the figure below.

Figure 110: IX with eBGP Route Server

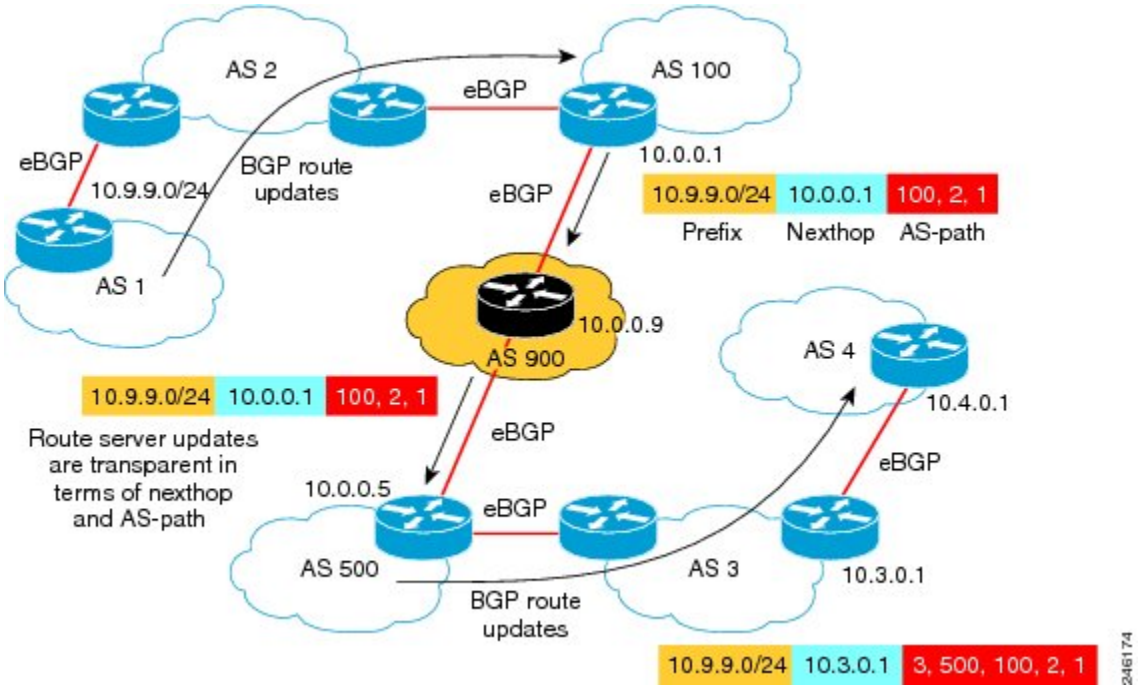


Instead of maintaining individual, direct eBGP peerings with every other provider, an SP maintains only a single connection to the route server operated by the IX. Peering with only the route server reduces the configuration complexity on each border router, reduces CPU and memory requirements on the border routers, and avoids most of the operational overhead incurred by individualized peering agreements.

The route server provides AS-path, MED, and nexthop transparency so that peering SPs at the IX still appear to be directly connected. In reality, the IX route server mediates this peering, but that relationship is invisible outside of the IX.

The figure below illustrates an example of transparent route propagation with a route server at an IX.

Figure 111: Transparent Route Propagation with Route Server at IX



In the figure above, a routing update goes from AS 1 to AS 2 to AS 100. The update leaves the router in AS 100 advertising that the router can reach the prefix 10.9.9.0/24, use 10.0.0.1 as the next hop, and use the AS path of AS100, AS2, AS1.

The router in AS 900 is a route server and the router in AS 500 is a route server client. A route server client receives updates from a route server. As shown in the figure above, the router in AS 900 does not change the update; route server updates are transparent in terms of MED, next hop and AS-path. The update goes to the client with the same prefix, next hop and AS-path that came from the router at 10.0.0.1.

Benefits of a BGP Route Server

A BGP route server provides the following benefits:

- Reduced configuration complexity on each border router.
- Reduced CPU and memory requirements on each border router.
- Reduced operational overhead incurred by individualized peering agreements.
- The ability for a route server context to override the normal BGP best path with an alternative path based on some policy.
- The ability for a route server context to suppress all paths for a prefix and therefore not advertise the prefix.

Route Server Context Provides Flexible Routing Policy

A BGP route server can provide a flexible routing policy. Your network environment might or might not have routes that need a customized (flexible) policy handling.

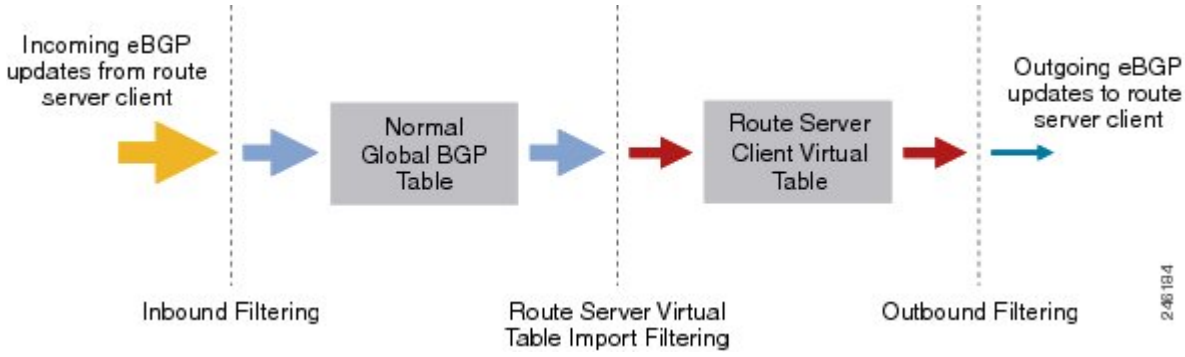
Routes needing flexible policy handling are selected for import into a route server context by configuring an import map. The import map references a route map, where the actual policy is defined by at least one **permit** statement. Routes (paths) that match the route map are included in a second "best path" calculation. The resulting best path, if it is different from the global best path, is imported into the context. Router server clients associated with a router server context will override the global best path with the context's best path when sending routing updates.

Multiple contexts can be created, and the appropriate context is referenced on the route server by the neighbors assigned to use that context (in the **neighbor route-server-client** command). Thus, multiple neighbors sharing the same policy can share the same route server context.

Three Stages of Filtering on a Route Server Client

With the introduction of route server context, there are now three stages of route filtering that can be applied to a route server client, as shown in the figure below. The three stages of filtering are described below the figure. You can apply all, none, or any combination of the three filtering methods to a route server client. In the figure, the decreasing arrow sizes symbolize that potentially fewer routes might pass each filter than entered the filter.

Figure 112: Route Server Filtering in Three Stages



1. As shown in the figure above beginning at the left, when incoming eBGP updates arrive from a route server client, the system will apply inbound route filters for a route server client the same way it does for a non-route-server client (configured with the **neighbor route-map in** command). All routes permitted by the client's inbound filtering are installed in the global BGP table for the appropriate address family, as usual, and anything else is dropped.
2. If any route server contexts have been configured with flexible policy using the **import-map** command, the best path from among the subset of matching routes is imported into the virtual table for the contexts. Route server clients associated with a context will then override any routes from the global BGP table with customized routes from the context's virtual table when generating updates.
3. A route server client's outbound filtering policies (configured with the **neighbor route-map out** command) will be applied to the global updates that do not have customized policy, and the outbound filtering policies are also applied to any updates generated from the route server context's virtual table.

How to Configure a BGP Route Server

Configure a Route Server with Basic Functionality

Perform this task to configure a BGP router as route server for IPv4 or IPv6. This task will enable the basic route server functionality for nexthop, AS-path, and MED transparency.

Note This task does not enable flexible policy handling. To enable flexible policy handling, see the Configure a Route Server with Flexible Policy Handling, on page 1334.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number*
5. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
6. **neighbor** {*ipv4-address*| *ipv6-address*} **activate**
7. **neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 4	**neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number* **Example:**	Adds an entry to the BGP neighbor table.

	Command or Action	Purpose
	`Router(config-router)# neighbor 10.0.0.1 remote-as 100`	
Step 5	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.
Step 6	**neighbor** {*ipv4-address*| *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 7	**neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 route-server-client`	Configures the BGP neighbor at the specified address to be a route server client.
Step 8	**end** **Example:** `Router(config-router-af)# end`	Ends the current configuration and returns to privileged EXEC mode.

Configure a Route Server Client To Receive Updates

In the prior task, you configured a route server. A route server does not put its own AS number in the AS-path; there is AS-path transparency. This means the route server client will receive updates in which the first AS number in the AS-path is not the sending router's AS number.

By default, a router denies an update received from an eBGP peer that does not list its AS number at the beginning of the AS-path in an incoming update. Therefore, you must disable that behavior on the client in order for the client to receive the updates. To do so, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp enforce-first-as**
5. **neighbor** {*ipv4-address*\| *ipv6-address*} **remote-as** *remote-as-number*
6. **address-family** {**ipv4** \| **ipv6**} { **unicast** \| **multicast**}
7. **neighbor** {*ipv4-address*\| *ipv6-address*} **activate**
8. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 4	**no bgp enforce-first-as** **Example:** `Router(config-router)# no bgp enforce-first-as`	Disables requirement that an update received from an eBGP peer list its AS number at the beginning of the AS_PATH. • By default, a router is configured to deny an update received from an external BGP (eBGP) peer that does not list its autonomous system number at the beginning of the AS_PATH in the incoming update. • In order to receive updates from the route server, which will not have its AS first in the AS_PATH, specify **no bgp enforce-first-as**to disable the enforcement.
Step 5	**neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 100`	Adds an entry to the BGP neighbor table.
Step 6	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.
Step 7	**neighbor** {*ipv4-address*| *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**exit-address-family** **Example:**	Exits address family configuration mode.

	Command or Action	Purpose
	`Router(config-router-af)# exit-address-family`	

Configure a Route Server with Flexible Policy Handling

Perform this task if you need your BGP route server to support a customized, flexible policy in addition to basic route server functionality.

In order to configure flexible policy handling, create a route server context, which includes an import map. The import map references a standard route map.

In this particular configuration task, the policy is based on autonomous system number, so the **match as-path** command is used. The actual AS number is identified in the **ip as-path access-list** command. You may match on nexthop, AS path, communities, and extended communities.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **route-server-context** *context-name*
5. **description** *string*
6. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
7. **import-map** *route-map-name*
8. **exit-address-family**
9. **exit-route-server-context**
10. **exit**
11. **ip as-path access-list** *access-list-number* {**permit**| **deny**} *regexp*
12. **route-map** *route-map-name* [**permit** | **deny**] *sequence-number*
13. **match as-path** *access-list-number*
14. **exit**
15. **router bgp** *autonomous-system-number*
16. **neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number*
17. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
18. **neighbor** {*ipv4-address*| *ipv6-address*} **activate**
19. **neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client context** *ctx-name*
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 4	**route-server-context** *context-name* **Example:** `Router(config-router)# route-server-context ONLY_AS27_CONTEXT`	Creates a route server context. • In this example, a context named ONLY_AS27_CONTEXT is created.
Step 5	**description** *string* **Example:** `Router(config-router-rsctx)# description Permit only routes with AS 27 in AS path.`	(Optional) Allows you to describe the context. • Up to 80 characters are allowed.
Step 6	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router-rsctx)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.
Step 7	**import-map** *route-map-name* **Example:** `Router(config-router-rsctx-af)# import-map only_AS27_routemap`	Configures flexible policy handling by using the route map that you will create in Step 12 to control which routes will be added to the route server client virtual table.
Step 8	**exit-address-family** **Example:** `Router(config-router-rsctx-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**exit-route-server-context** **Example:** `Router(config-router-rsctx)# exit-route-server-context`	Exits route server context configuration mode.
Step 10	**exit** **Example:**	Exits router configuration mode.

	Command or Action	Purpose
	`Router(config-router)# exit`	
Step 11	**ip as-path access-list** *access-list-number* {**permit**| **deny**} *regexp* **Example:** `Router(config)# ip as-path access-list 5 permit 27`	Configures an AS path filter using a regular expression. • The **ip as-path** command is not necessarily the command you have to use. Determine what policy you want to create.
Step 12	**route-map** *route-map-name* [**permit** | **deny**] *sequence-number* **Example:** `Router(config)# route-map only_AS27_routemap permit 10`	Defines whether AS paths that match the subsequent **match as-path**command will be permitted or denied in the route map. • Use the same *route-map-name* that you specified in the **import-map** command above.
Step 13	**match as-path** *access-list-number* **Example:** `Router(config-route-map)# match as-path 5`	Identifies an access list that determines which AS paths are matched and become part of the route map configured in the prior step. • This particular example references the *access-list-number* configured in the **ip as-path access-list** command. • The **match as-path** command is not necessarily the command you have to use. Determine what policy you want to use. • You may match on nexthop, AS path, communities, and extended communities.
Step 14	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode.
Step 15	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 900`	Configures a BGP routing process.
Step 16	**neighbor** {*ipv4-address*| *ipv6-address*} **remote-as** *remote-as-number* **Example:** `Router(config-router)# neighbor 10.0.0.1 remote-as 500`	Adds an entry to the BGP neighbor table.

	Command or Action	Purpose
Step 17	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure a routing session using IPv4 or IPv6 unicast or multicast address prefixes.
Step 18	**neighbor** {*ipv4-address*| *ipv6-address*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 19	**neighbor** {*ipv4-address*| *ipv6-address*} **route-server-client context** *ctx-name* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 route-server-client context ONLY_AS27_CONTEXT`	Configures the BGP neighbor at the specified address to be a route server client. • In this example, the route server client at this specified address is assigned to the context called ONLY_AS27_CONTEXT.
Step 20	**end** **Example:** `Router(config-router-af)# end`	Ends the current configuration and returns to privileged EXEC mode.

Displaying BGP Route Server Information and Troubleshooting Route Server

From privileged EXEC mode, perform either of the steps in this task on a BGP route server to see information about the route server.

On a BGP route server client (not the route server), you can use the **show ip bgp ipv4 unicast** or **show ip bgp ipv6 unicast**command to display routing information.

SUMMARY STEPS

1. **enable**
2. **show ip bgp** {**ipv4** | **ipv6**} **unicast route-server** {**all** | {**context** *context-name*}} [**summary**]
3. **debug ip bgp route-server** {**client** | **context** | **event** | **import** | **policy**} [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip bgp** {**ipv4** | **ipv6**} **unicast route-server** {**all** | {**context** *context-name*}} [**summary**] **Example:** `Router#`	Displays the paths chosen for a particular route server context, which migh include the global bestpath, an overriding policy path, or a suppressed path.
Step 3	**debug ip bgp route-server** {**client** | **context** | **event** | **import** | **policy**} [**detail**] **Example:** `Router# debug ip bgp route-server client`	Turns on debugging for BGP route server. **Caution** The **detail** keyword is used for more complex issues and should only be turned on when debugging with a Cisco representative.

Configuration Examples for BGP Route Server

Example BGP Route Server with Basic Functionality

In the following example, the neighbor at 10.0.0.1 is a route server client.

```
router bgp 65000
 neighbor 10.0.0.1 remote-as 100
 neighbor 10.0.0.5 remote-as 500
 address-family ipv4 unicast
 neighbor 10.0.0.1 activate
 neighbor 10.0.0.1 route-server-client
!
```

Example BGP Route Server Context for Flexible Policy (IPv4 Addressing)

In the following example, the local router is a BGP route server. Its neighbors at 10.10.10.12 and 10.10.10.13 are its route server clients. A route server context named ONLY_AS27_CONTEXT is created and applied to the neighbor at 10.10.10.13. The context uses an import map that references a route map named only_AS27_routemap. The route map matches routes permitted by access list 27. Access list 27 permits routes that have 27 in the AS path.

```
router bgp 65000
   route-server-context ONLY_AS27_CONTEXT
      address-family ipv4 unicast
         import-map only_AS27_routemap
      exit-address-family
   exit-route-server-context
   !
   neighbor 10.10.10.12 remote-as 12
   neighbor 10.10.10.12 description Peer12
   neighbor 10.10.10.13 remote-as 13
   neighbor 10.10.10.13 description Peer13
   neighbor 10.10.10.21 remote-as 21
   neighbor 10.10.10.27 remote-as 27
   !
   address-family ipv4
```

```
      neighbor 10.10.10.12 activate
      neighbor 10.10.10.12 route-server-client
      neighbor 10.10.10.13 activate
      neighbor 10.10.10.13 route-server-client context ONLY_AS27_CONTEXT
      neighbor 10.10.10.21 activate
      neighbor 10.10.10.27 activate
   exit-address-family
!
ip as-path access-list 27 permit 27
!
route-map only_AS27_routemap permit 10
   match as-path 27
!
```

Example Using Show Commands to See That Route Server Context Routes Overwrite Normal Bestpath

In the following output, a BGP route server has two routes from AS 21 that have been selected as best:

```
Route-Server# show ip bgp ipv4 unicast
BGP table version is 31, local router ID is 100.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.1/32       10.10.10.21             23             0 21 ?
*                   10.10.10.27            878             0 27 89 ?
*  100.1.1.1/32     10.10.10.27            878             0 27 89 ?
*>                  10.10.10.21             23             0 21 ?
```

For Peer12, which has been configured as a route-server client, but not associated with any context, the bestpath is advertised in the following output. Note that AS-path, MED, and nexthop transparency have been maintained; the routes look as if they had not passed through the route server.

```
Peer12# show ip bgp ipv4 unicast

BGP table version is 31, local router ID is 10.10.10.12
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.1/32       10.10.10.21             23             0 21 ?
*> 100.1.1.1/32     10.10.10.21             23             0 21 ?
```

Peer13 has also been configured as a route-server client, and it has been associated with a context named ONLY_AS27_CONTEXT. The context references a route map that permits only routes that contain AS 27 in the AS path. This means that the route-server should not send any routes to Peer13 unless they contains AS 27. In our scenario, the route server indeed sends the routes learned via AS 27, even though the routes learned via AS 21 are marked as best. The output below demonstrates that the normal best path was overriden by the best path based on policy. Again, MED, as-path, and nexthop transparency have been maintained.

```
Peer13# show ip bgp ipv4 unicast

BGP table version is 25, local router ID is 10.10.10.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
```

```
*> 1.1.1.1/32       10.10.10.27            878             0 27 89 ?
*> 100.1.1.1/32     10.10.10.27            878             0 27 89 ?
```

Example BGP Route Server Context with No Routes Satisfying the Policy

It is possible that, due to policy, no routes are sent to a client even though paths exist. For instance, if we take the prior example and change ONLY_AS27_CONTEXT to ONLY_AS100_CONTEXT, no paths would satisfy this policy and no routes will be sent to the client. The following is the configuration and resulting show output:

```
Route-Server# show run | begin router bgp
router bgp 1
   route-server-context ONLY_AS100_CONTEXT
     !
     address-family ipv4 unicast
        import-map only_AS100_routemap
     exit-address-family
   exit-route-server-context
   !
   neighbor 10.10.10.13 remote-as 13
   neighbor 10.10.10.13 description Peer13
   neighbor 10.10.10.21 remote-as 21
   neighbor 10.10.10.27 remote-as 27
   !
      address-family ipv4
         neighbor 10.10.10.13 activate
         neighbor 10.10.10.13 route-server-client context ONLY_AS100_CONTEXT
         neighbor 10.10.10.21 activate
         neighbor 10.10.10.27 activate
   exit-address-family

!
ip as-path access-list 100 permit 100
!
!
route-map only_AS100_routemap permit 10
   match as-path 100
!
```

Because no routes satisfy the policy, no routes appear in the table of Peer13:

```
Peer13# show ip bgp ipv4 unicast
```

Example BGP Route Server Context for Flexible Policy (IPv6 Addressing)

In the following example under address-family IPv6, the local router is a BGP route server. Its neighbors at 2001:DB8:1::112 and 2001:DB8:1::113 are its route server clients. A route server context named ONLY_AS27_CONTEXT is created and applied to the neighbor at 2001:DB8:1::113. The context uses an import map that references a route map named only_AS27_routemap. The route map matches routes permitted by access list 27. Access list 27 permits routes that have 27 in the AS path.

```
Route-Server# show run | begin router bgp
router bgp 1
   route-server-context ONLY_AS27_CONTEXT
      address-family ipv6 unicast
         import-map only_AS27_routemap
      exit-address-family
   exit-route-server-context
```

```
   !
   neighbor 2001:DB8:1::112 remote-as 12
   neighbor 2001:DB8:1::112 description Peer12
   neighbor 2001:DB8:1::113 remote-as 13
   neighbor 2001:DB8:1::113 description Peer13
 !
   address-family ipv6
      neighbor 2001:DB8:1::112 activate
      neighbor 2001:DB8:1::112 route-server-client
      neighbor 2001:DB8:1::113 activate
      neighbor 2001:DB8:1::113 route-server-client context ONLY_AS27_CONTEXT
 exit-address-family
!
ip as-path access-list 27 permit 27
!
route-map only_AS27_routemap permit 10
   match as-path 27
!
Route-Server#show ip bgp ipv6 unicast route-server all summary

Route server clients without assigned contexts:
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:DB8:1::112 4           12      19      19        4    0    0 00:12:50        2
Route server clients assigned to context ONLY_AS27_CONTEXT:
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:DB8:1::113 4           13      23      22        4    0    0 00:16:23        2
```

For Peer12, which has been configured as a route-server client, but not associated with any context, the bestpath is advertised. Note that AS-path, MED, and nexthop transparency have been maintained; the routes look as if they had not passed through the route server.

```
Peer12# show ip bgp ipv6 unicast
BGP table version is 9, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
* 2001:DB8:1::/64  2001:DB8::113            0             0 13 ?
*>                  ::                       0         32768 ?
* 2001:DB8:2::/64  2001:DB8::113            0             0 13 ?
*>                  ::                       0         32768 ?
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	*Cisco IOS IP Routing: BGP Command Reference*
BGP configuration tasks	*IP Routing: BGP Configuration Guide, Cisco IOS XE Release 3S*

MIBs

MIB	MIBs Link
-	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

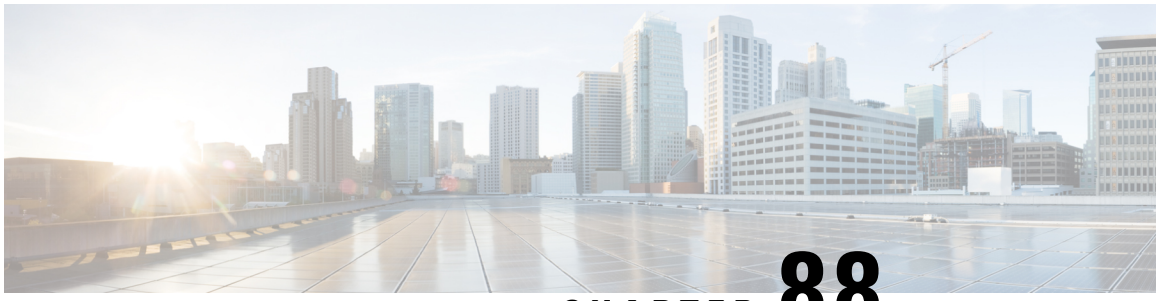
Feature Information for BGP Route Server

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 113: Feature Information for BGP Route Server

Feature Name	Releases	Feature Information
BGP Route Server	Cisco IOS XE Release 3.3S 15.2(3)T	BGP route server is a feature designed for internet exchange (IX) operators that provides an alternative to full eBGP mesh peering among the service providers who have a presence at the IX. The route server provides eBGP route reflection with customized policy support for each service provider. That is, a route server context can override the normal BGP best path for a prefix with a different path based on a policy, or suppress all paths for a prefix and not advertise the prefix. The BGP route server provides reduced configuration complexity and reduced CPU and memory requirements on each border router. The route server also reduces overhead expense incurred by individualized peering agreements. The following commands were introduced: • **debug ip bgp route-server** • **description** (route server context) • **exit-route-server-context** • **import-map** • **neighbor route-server-client** • **route-server-context** • **show ip bgp unicast route-server**



CHAPTER 88

BGP Diverse Path Using a Diverse-Path Route Reflector

The BGP Diverse Path Using a Diverse-Path Route Reflector feature allows Border Gateway Protocol (BGP) to distribute an alternative path other than the best path between BGP speakers when route reflectors are deployed. This feature is meant to provide path diversity within an autonomous system (AS), within a single cluster only. That is, a route reflector is allowed to advertise the diverse path to its client peers only.

Prerequisites for BGP Diverse Path Using a Diverse-Path Route Reflector

You should understand the BGP Best External feature.

Restrictions for BGP Diverse Path Using a Diverse-Path Route Reflector

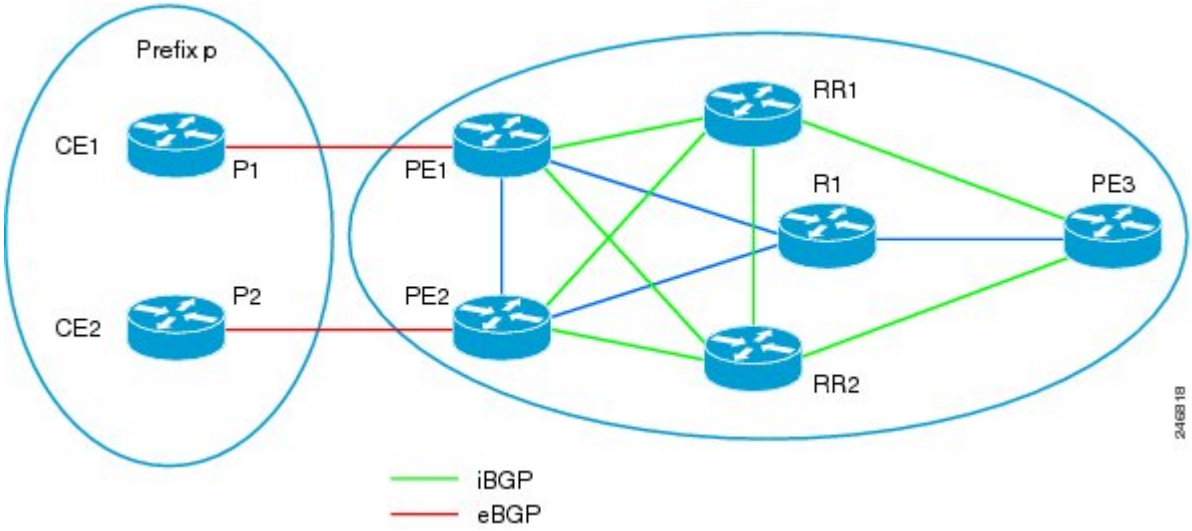
- A diverse path can be configured on a route reflector only.
- Only one shadow route reflector is allowed per existing route reflector, which will calculate one additional best path (the second best path). That is, only one additional plane (topology) is configured.
- Path diversity is configured within an AS, within a single route reflector cluster. That is, the route reflector will advertise the diverse path to its route reflector client peers only.
- Diverse path functionality is not supported on a route server.

Information About BGP Diverse Path Using a Diverse-Path Reflector

Limitation that a BGP Diverse Path Overcomes

As a path vector routing protocol, BGP-4 requires a router to advertise to its neighbors only the best path for a destination. However, multiple paths to the same destination would allow mechanisms that can improve resilience, quickly recover from failures, and load balance, for example.

The use of route reflectors is one of the main reasons for poor path diversity within an autonomous system (AS). In a network with route reflectors, even if a prefix is learned from multiple egress routers, the route reflector reflects only the best path to its clients. The figure below shows how deploying route reflectors might reduce path diversity in an AS, even when the BGP Best External feature is deployed.



In the figure above, P1 and P2 are diverse paths for prefix p. Assume Path 2 (P2) has a lower MED and higher local preference than P1. The BGP Best External feature on PE1 will make sure that P1 is propagated to the route reflectors, regardless of P2 having a lower MED and higher local preference. The route reflectors will have path diversity; they will learn both P1 and P2 with different exit points PE1 and PE2 (assuming that PE1 and PE2 have the **set ip next-hop self** command configured). However, both route reflectors select the best path as P2 due to its lower MED/higher local preference and advertise it to PE3. PE3 will not learn P1 (that is, PE3 will not learn about existing path diversity).

The BGP Diverse Path Using a Diverse-Path Route Reflector feature is a way to resolve that limitation and achieve path diversity.

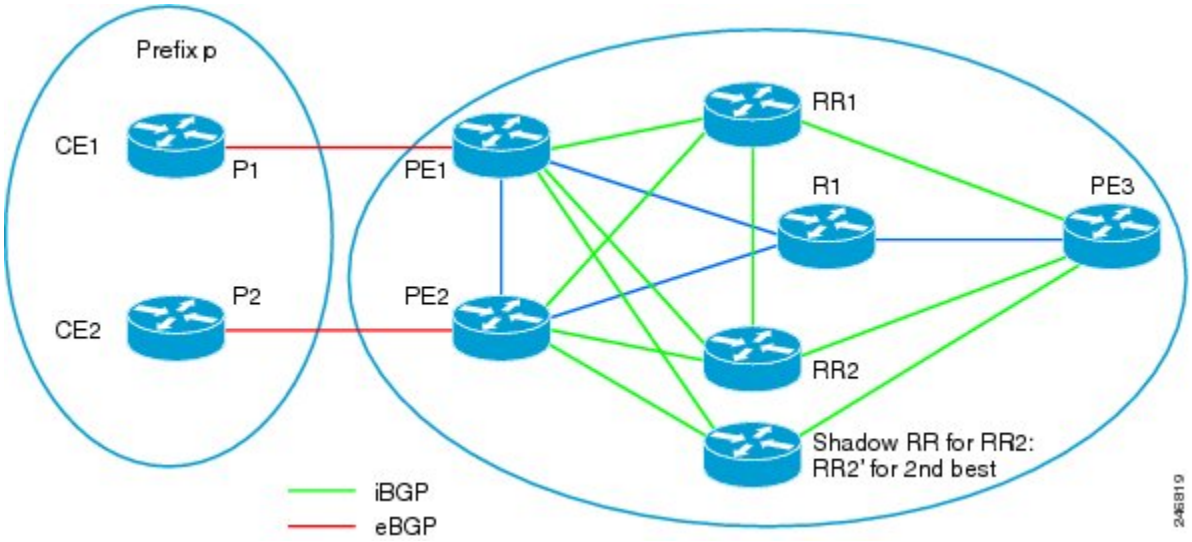
BGP Diverse Path Using a Diverse-Path Route Reflector

The BGP Diverse Path Using a Diverse-Path Route Reflector feature overcomes the lack of path diversity in an AS containing route reflectors. This feature is meant to provide path diversity within an AS, within a single cluster only. That is, a route reflector is allowed to advertise the diverse path to its client peers only.

For each route reflector in the AS, a *shadow route reflector* is added to distribute the *second best path*, also known as the *diverse path*. The figure below shows the shadow route reflector for RR2. The shadow route reflector improves path diversity because PE3 can now learn both P1 (from RR1/RR2) and learn P2 from the shadow route reflector.

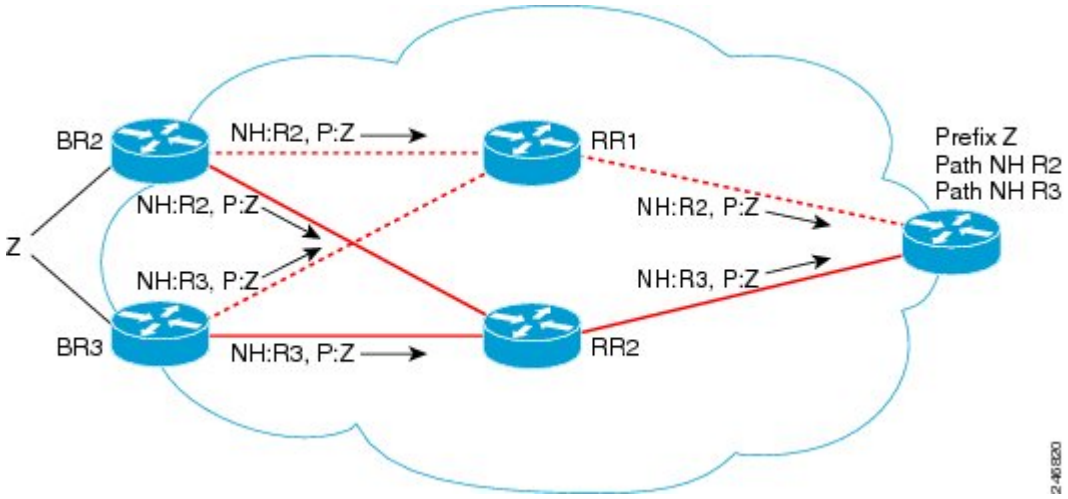
Note The primary route reflector and shadow route reflector must have the exact same connections (physical/control plane) to the rest of the routers in the network.

Shadow route reflectors can be both control plane route reflectors and data plane route reflectors.



The figure below shows a diverse path in greater detail, indicating the next hops:

- BR2 announces to RR1 and shadow RR2 that R2 (BR2) is the Next Hop for those who want to reach Prefix Z. Likewise, BR3 announces to RR1 and shadow RR2 that R3 (BR3) is the Next Hop for those who want to reach Prefix Z
- RR1 sends a packet to BR1 announcing that the Next Hop is R2 if BR1 wants to reach Prefix Z. The second best path (or diverse path) comes from shadow RR2, which sends a packet to BR1 announcing that the Next Hop is R3 if BR1 want to reach Prefix Z.
- At BR1 (far right), we see there are two (diverse) paths to Prefix Z.



Triggers to Compute a BGP Diverse Path

Computation of a diverse path per address family is triggered by any of the following commands:

- **bgp additional-paths install**
- **bgp additional-paths select**
- **maximum-paths ebgp**
- **maximum-paths ibgp**

The **bgp additional-paths install** command will install the type of path that is specified in the **bgp additional-paths select** command. If the **bgp additional-paths select** command specifies both keyword options (**best-external** and **backup**), the system will install a backup path.

The **maximum-paths ebgp** and **maximum-paths ibgp**commands trigger a multipath computation, and multipaths are automatically installed as primary paths.

On the other hand, the **bgp additional-paths install** command triggers computation of a backup path or best-external path.

If the **bgp additional-paths select** command is not configured, the **bgp additional-paths install**command will trigger both computation and installation of a backup path (as is done with the BGP PIC feature).

IGP Metric Check

Disabling the Interior Gateway Protocol (IGP) metric check and configuring the BGP Diverse Path feature are independent of each other. One does not imply the other. That is, configuring **bgp bestpath igp-metric ignore** does not imply that the BGP Diverse Path feature is enabled. Conversely, enabling the BGP Diverse Path feature might not require that **bgp bestpath igp-metric ignore** be configured (because, for example, the route reflector and shadow route reflector are co-located).

The **bgp bestpath igp-metric ignore** command can be configured at route reflectors and provider edges (PEs).

Note Per-VRF functionality for the **bgp bestpath igp-metric ignore** command is not supported. If you use it anyway, it is at your own risk.

Route Reflector Determination

If a router's configuration includes either one of the following commands, the router is a route reflector:

- **bgp cluster-id**
- **neighbor route-reflector-client**

How to Configure a BGP Diverse-Path Route Reflector

Determining Whether You Need to Disable the IGP Metric Check

Before you configure a shadow route reflector in order to get a BGP diverse path, determine whether you need to disable the IGP metric check. The IGP metric is a configurable value indicating physical distance, and is used by an Interior Gateway Protocol, such as Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), or Routing Information Protocol (RIP). A smaller IGP metric is preferred over a larger IGP metric.

The locations of the route reflector and shadow route reflector determine whether or not you need to disable the IGP metric check, as follows:

- When the route reflector and shadow route reflector are colocated—They have the same IP subnetwork address and are connected to the Ethernet switch with different links. Failure of such a link is equivalent to the route reflector going down. When RRs are colocated, their IGP metrics cannot be different from each other; and therefore there is no need to disable the IGP metric check during the best path calculation at any route reflector. Because there is no need to disable the IGP metric check, the first plane route reflectors do not need to be upgraded to Cisco IOS XE Release 3.4S.
- When the shadow route reflector is in a different IGP place from the route reflector (it is not colocated with its best path route reflector)--In this case, the IGP metric check is ignored on both the best path route reflector and shadow route reflector when the best path and second best path are being calculated. The IGP metric check must be disabled on the primary route reflector by configuring the **bgp bestpath igp-metric ignore** command. This command is available beginning with Cisco IOS XE Release 3.4S, which means you need to upgrade to that release.

Configuring the Route Reflector for BGP Diverse Path

Perform this task to configure a route reflector for the BGP Diverse Path feature. This task specifies the IPv4 address family, but other address families are also supported.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4 unicast**
6. **neighbor** *ip-address* **activate**
7. **maximum-paths ibgp** *number-of-paths*
8. **bgp bestpath igp-metric ignore**
9. **bgp additional-paths select** [**backup**]
10. **bgp additional-paths install**
11. **neighbor** *ip-address* **route-reflector-client**
12. **neighbor** *ip-address* **advertise diverse-path** [**backup**] [**mpath**]
13. **end**
14. **show ip bgp neighbor** *ip-address* **advertised-routes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 1`	Enters router configuration mode for the BGP routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.1.1.1 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 5	**address-family ipv4 unicast** **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the address family and enters address family configuration mode. • Supported address families are IPv4 unicast, VPNv4 unicast, IPv6 unicast, VPNv6 unicast, IPv4+label, and IPv6+label.
Step 6	**neighbor** *ip-address* **activate** **Example:**	Enables the exchange of information with a BGP neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 10.1.1.1 activate`	
Step 7	**maximum-paths ibgp** *number-of-paths* **Example:** `Device(config-router-af)# maximum-paths ibgp 4`	Controls the maximum number of parallel Internal BGP (IBGP) routes that can be installed in a routing table.
Step 8	**bgp bestpath igp-metric ignore** **Example:** `Device(config-router-af)# bgp bestpath igp-metric ignore`	Configures the system to ignore the Interior Gateway Protocol (IGP) metric during BGP best path selection.
Step 9	**bgp additional-paths select** [**backup**] **Example:** `Device(config-router-af)# bgp additional-paths select backup`	Configures the system to calculate a second BGP best path.
Step 10	**bgp additional-paths install** **Example:** `Device(config-router-af)# bgp additional-paths install`	Enables BGP to calculate a backup path for a given address family and to install it into the routing information base (RIB) and Cisco Express Forwarding (CEF).
Step 11	**neighbor** *ip-address* **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 10.1.1.1 route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client.
Step 12	**neighbor** *ip-address* **advertise diverse-path** [**backup**] [**mpath**] **Example:** `Device(config-router-af)# neighbor 10.1.1.1 advertise diverse-path backup`	(Optional) Configures a neighbor to receive the diverse path in an advertisement.
Step 13	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits address family configuration mode and returns to privileged EXEC mode.
Step 14	**show ip bgp neighbor** *ip-address* **advertised-routes** **Example:** `Device# show ip bgp neighbor 10.1.1.1 advertised-routes`	(Optional) Displays the routes advertised to the specified neighbor.

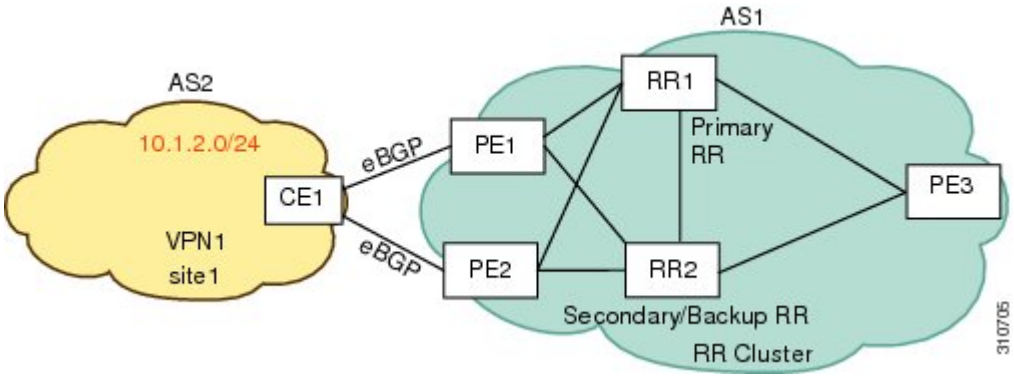
Configuration Examples for BGP Diverse Path Using a Diverse-Path Route Reflector

Example: Configuring BGP Diverse Path Where Additional Path Is the Backup Path

Diverse path functionality is contained within a single cluster; that is, only the clients of a route reflector can be configured to advertise the diverse path. A diverse path is advertised to the clients of a route reflector only if the client is configured to get the additional path.

A shadow route reflector can be added to calculate and advertise the additional path, or an existing route reflector can be configured to calculate and advertise the additional path. In the figure below, instead of adding a shadow route reflector, RR2 (the existing backup RR) is configured to calculate the additional path and advertise it to a particular neighbor.

In the figure below, assume that from the route reflectors, the path to CE1 via PE1 is preferred over the path via PE2. Without the diverse path feature, both route reflectors will advertise to PE3 that the path to CE1 is via PE1. If the connection between RR1 and PE1 fails (or the path between PE1 and CE1 fails), there is no other path.



In the following configuration example based on the figure above, RR2 is configured with an additional path, which is a backup path.

If RR1 and RR2 are not colocated, you must configure the **bgp bestpath igp-metric ignore** command before the additional path is calculated. (If RR1 and RR2 are colocated, do not configure that command.)

The **bgp additional-paths select backup** command triggers calculation of the backup path at RR2, which is the path via PE2.

The **bgp additional-paths install** command installs the backup path if RR2 is in the forwarding plane. (Do not configure this command if RR2 is in the control plane.)

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path backup** command on RR2. This command triggers advertisement of the backup path to PE3. PE3 will learn the best path, (which is the path via PE1) from RR1, and it will learn the backup path from RR2.

RR2

```
router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 bgp bestpath igp-metric ignore
 bgp additional-paths select backup
 bgp additional-paths install
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path backup
```

Example: Configuring BGP Diverse Path Where Additional Path Is the Multipath

In the following example based on the figure above, assume that paths toward CE1 via PE1 and PE2 are multipaths. The **maximum-paths ibgp** command will trigger calculation of multipaths.

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path mpath** command on RR2. This command will trigger advertisement of the multipath, that is, the second best path, to PE3. PE3 will learn the best path, path via PE1 from RR1, and will learn second best path from RR2.

RR2

```
router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path mpath
```

Example: Configuring BGP Diverse Path Where Both Multipath and Backup Path Calculations Are Triggered

The following example is based on the figure above. The **maximum-paths ibgp** command will trigger calculation of multipaths. When both multipath and backup path calculations are triggered, the backup path and the second multipath (which is the second best path) are the same paths and it will be installed as the active path, regardless of whether the route reflector is in the control plane or forwarding plane.

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path backup mpath** command on RR2. This command causes RR2 to advertise the second best path, which is the second multipath, to PE3.

RR2

```
router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 bgp additional-paths select backup
 neighbor 10.1.1.1 remote-as 1
```

```
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path backup mpath
```

Example: Configuring Triggering Computation and Installation of a Backup Path

When the **bgp additional-paths install** command is configured without configuring **bgp additional-paths select backup**, the former command will trigger both computation and installation of the backup path (as it is with the existing BGP PIC feature).

The address of PE3 is 10.1.1.1, and that address is used in the **neighbor advertise diverse-path backup** command on RR2. This command will trigger advertisement of a backup path to PE3. PE3 will learn the best path, a path via PE1 from RR1, and it will learn a backup path from RR2.

RR2

```
router bgp 1
 neighbor 10.1.1.1 remote-as 1
 address-family ipv4 unicast
 neighbor 10.1.1.1 activate
 maximum-paths ibgp 4
 bgp additional-paths install
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 route-reflector-client
 neighbor 10.1.1.1 advertise diverse-path backup
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Configuring BGP Best External Path on a Route Reflector for Intercluster	BGP Best External module
BGP configuration tasks	Cisco IOS XE IP Routing: BGP Configuration Guide

Standards

Standard	Title
draft-ietf-grow-diverse-bgp-path-dist-02.txt	*Distribution of Diverse BGP Paths*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 4271	A Border Gateway Protocol 4 (BGP-4)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

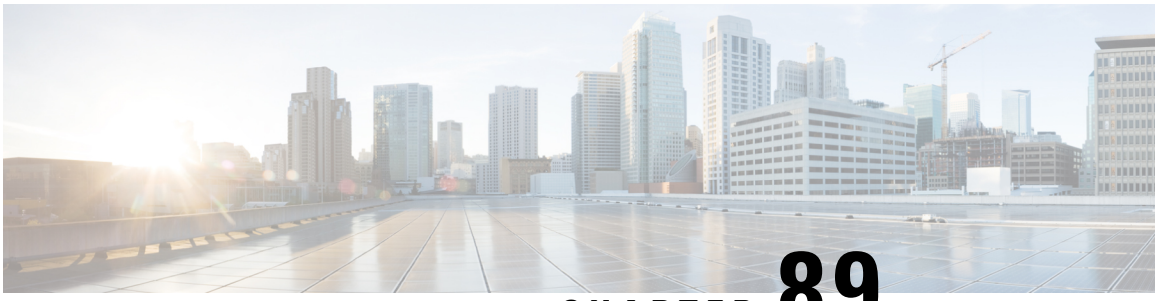
Feature Information for BGP Diverse Path Using a Diverse-Path Route Reflector

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 114: Feature Information for BGP Diverse Path Using a Diverse-Path Route Reflector

Feature Name	Releases	Feature Information
BGP Diverse Path Using a Diverse-Path Route Reflector		This feature allows BGP to distribute an alternative path other than the best path between BGP speakers when route reflectors are deployed. The following commands were introduced: • **bgp additional-paths select** • **bgp bestpath igp-metric ignore** • **debug ip bgp igp-metric ignore** • **neighbor advertise best-external** • **neighbor advertise diverse-path**



CHAPTER 89

BGP Enhanced Route Refresh

The BGP Enhanced Route Refresh feature provides a way for Border Gateway Protocol (BGP) to find route inconsistencies, and in that unlikely event, to synchronize BGP peers without a hard reset. The feature is enabled by default; there are two optional timers.

Information About BGP Enhanced Route Refresh

BGP Enhanced Route Refresh Functionality

During session establishment, BGP peers exchange with each other their capability to do the BGP Enhanced Route Refresh feature. The feature is enabled by default.

It is not expected that the peers will become inconsistent with each other. That might only happen in an extreme corner case, and if that happens, this feature helps to identify that and synchronize the peers without a hard reset.

If two peers are capable of Enhanced Route Refresh, each peer will generate a Route-Refresh Start-of-RIB (SOR) message before it advertises the Adj-RIB-Out, and will generate a Route-Refresh End-of-RIB (EOR) message after it advertises the Adj-RIB-Out. A BGP speaker receiving an EOR message from its peer removes the routes that were not re-advertised as part of Route Refresh response by the peer.

In the unlikely event the router has stale routes remaining after receiving the EOR message or after the EOR timer expires, that means the peers were not consistent with each other. This information can be used to check whether routes are consistent.

BGP Enhanced Route Refresh Timers

These timers need not be configured under normal circumstances.You could configure one or both timers if you observe there is continuous route flapping to the extent that a Route Refresh EOR cannot be generated.

The first timer applies to the router when it should be receiving the EOR message, but is not receiving one. The second timer applies to the router when it should be sending the EOR message.

- Stale path timer—If the **bgp refresh stalepath-time** command is configured and the router does not receive a Route-Refresh EOR message after an Adj-RIB-Out, the router removes the stale routes from the BGP table after the timer expires. The stale path timer is started when the router receives a Route-Refresh SOR message.
- Maximum EOR timer—If the **bgp refresh max-eor-time** command is configured and the router is unable to generate a Route-Refresh EOR message, a Route-Refresh EOR message is generated after the timer expires.

Both timers are configurable. By default, they are both disabled (set to 0 seconds).

Syslog Messages Generated by the BGP Enhanced Route Refresh

The following are examples of syslog messages that are generated when a peer deletes stale routes after receiving the Route-Refresh EOR message or after the stale path timer expires. The messages help you to know whether the routers were inconsistent.

```
Net 300:300:3.3.0.0/0 from bgp neighbor IPv4 MDT 10.0.101.1 is stale after refresh EOR
(rate-limited)
Net 300:300:3.3.0.0/0 from bgp neighbor IPv4 MDT 10.0.101.1 is stale after refresh stale-path
 timer expiry (rate-limited)
```

The following are examples of messages logged after a Route-Refresh EOR or after the stale path timer expires, which indicate the total number of stale paths that were from the neighbor.

```
3 stale-paths deleted from bgp neighbor IPv4 MDT 10.0.101.1 after refresh EOR
3 stale-paths deleted from bgp neighbor IPv4 MDT 10.0.101.1 after refresh stale-path timer
 expiry
```

How to Set Timers for BGP Enhanced Route Refresh

Set Timers for BGP Enhanced Route Refresh

The BGP Enhanced Route Refresh feature is enabled by default; the timers are disabled by default. Perform this task if you want to set the optional timers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system*
4. **bgp refresh stalepath-time** *seconds*
5. **bgp refresh max-eor-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system* **Example:** `Router(config)# router bgp 65000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**bgp refresh stalepath-time** *seconds* **Example:** `Router(config-router)# bgp refresh stalepath-time 1200`	(Optional) Causes the router to remove stale routes from the BGP table after the timer expires, even if the router does not receive a Route-Refresh End-of-RIB message. • Valid values are from 600 to 3600, or 0. • The default is 0, meaning the stale-path timer is disabled. • The stale path timer is started when a router receives a Route-Refresh Start-of-RIB message.
Step 5	**bgp refresh max-eor-time** *seconds* **Example:** `Router(config-router)# bgp refresh max-eor-time 1200`	(Optional) Specifies that if BGP is unable to generate a Route-Refresh End-of-RIB (EOR) message, a Route-Refresh EOR is generated after the timer expires. • Valid values are from 600 to 3600, or 0. • The default is 0, meaning the max-eor timer is disabled.

Configuration Examples for BGP Enhanced Route Refresh

Example: Setting Timers for BGP Enhanced Route Refresh

In the following example, if no Route-Refresh EOR message is received after 800 seconds, stale routes will be removed from the BGP table. If no Route-Refresh EOR message is generated after 800 seconds, one is generated.

```
router bgp 65000
 bgp refresh stalepath-time 800
 bgp refresh max-eor-time 800
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Enhanced Route Refresh

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 115: Feature Information for BGP Enhanced Route Refresh

Feature Name	Releases	Feature Information
BGP Enhanced Route Refresh		The BGP Enhanced Route Refresh feature provides a way for BGP to find route inconsistencies, and in that unlikely event, to synchronize BGP peers without a hard reset. The following commands were introduced: • **bgp refresh max-eor-time** • **bgp refresh stalepath-time**

CHAPTER 90

Configuring BGP Consistency Checker

The BGP Consistency Checker feature provides a way to identify certain types of BGP route inconsistencies with peers: next-hop label inconsistency, RIB-out inconsistency, and aggregation inconsistency. Upon finding such an inconsistency, the system sends a syslog error message and takes appropriate action if configured to do so.

Information About BGP Consistency Checker

BGP Consistency Checker

A BGP route inconsistency with a peer occurs when an update or a withdraw is not sent to a peer resulting in a null route. To identify that issue, BGP consistency checker was created as a low-priority process that does nexthop-label, RIB-out, and aggregation consistency checks at a configurable interval. When enabled, BGP consistency checker is performed for all address families. Configuring BGP consistency checker is recommended.

Once the process identifies such an inconsistency, it will report the inconsistency with a syslog message and optionally take action if the **auto-repair** keyword is specified. The action taken depends on the type of inconsistency found.

- Next-Hop Label Consistency Check—When two paths have the same next hop because they are advertised by the same provider edge router (PE), they should also have the same next-hop label. If the labels are different, there is an inconsistency. If the **auto-repair** keyword is specified, the system will send a route-refresh request.
- RIB-Out Consistency Check—If a network passes an outbound policy and is not sent, or if a network does not pass an outbound policy and is sent, there is an inconsistency. If the **auto-repair** keyword is specified, the system will send a route-refresh request.
- Aggregation Consistency Check—If specific routes and the aggregated route become out of sync, an inconsistency can occur. Either the **error-message** keyword or the **auto-repair** keyword will trigger aggregation reevaluation.

In the unlikely event that you receive a syslog message about an inconsistency, notify your Cisco technical support representative with the syslog message exactly as it appears. The following are examples of such syslog messages:

- "Net 10.0.0.0/32 has Nexthop-Label inconsistency."
- "Net 10.0.0.0/32 in IPv4 Unicast has rib-out inconsistency for update-group 4 - outbound-policy fails."

How to Configure BGP Consistency Checker

Configure BGP Consistency Checker

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp consistency-checker** {**error-message** | **auto-repair**} [**interval** *minutes*]
5. **end**
6. **show ip bgp** [**vpnv4** | **vpnv6**] **all inconsistency nexthop-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 500`	Configures a BGP routing process.
Step 4	**bgp consistency-checker** {**error-message** | **auto-repair**} [**interval** *minutes*] **Example:** `Router(config-router)# bgp consistency-checker auto-repair interval 720`	Enables BGP consistency checker. • The default interval is 1440 minutes (one day). The range is 5 to 1440 minutes.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-router)# end`	Ends the current configuration and returns to privileged EXEC mode.
Step 6	**show ip bgp** [**vpnv4** | **vpnv6**] **all inconsistency nexthop-label** **Example:** `Router# show ip bgp all inconsistency nexthop-label`	(Optional) Displays routes that have a nexthop-label inconsistency found. • This step is not part of configuring the feature; it is provided in case you receive a syslog message about a nexthop-label inconsistency and you want to display those routes.

Configuration Examples for BGP Consistency Checker

Example: Configuring BGP Consistency Checker

The following example configures BGP consistency checker with auto-repair at the default interval of one day:

```
router bgp 65000
 bgp consistency-checker auto-repair
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Enabling BGP MIB support	"BGP MIB Support" module in the *IP Routing: BGP Configuration Guide*
Configuring SNMP Support	*SNMP Configuration Guide* in the *Cisco IOS Network Management Configuration Guide Library*
SNMP Commands	*Cisco IOS SNMP Support Command Reference*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1657	*BGP-4 MIB*
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 2547	*BGP/MPLS VPNs*
RFC 2858	*Multiprotocol Extensions for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

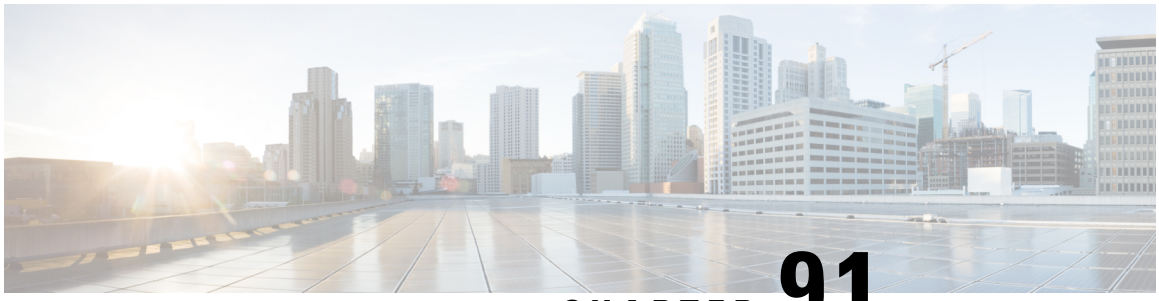
Feature Information for BGP Consistency Checker

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 116: Feature Information for BGP Consistency Checker

Feature Name	Releases	Feature Information
BGP Consistency Checker	Cisco IOS XE Release 3.3S Cisco IOS XE Release 3.4SG	The BGP Consistency Checker feature provides a way to identify three types of BGP route inconsistencies with peers: next-hop label inconsistency, RIB-out inconsistency, and aggregation inconsistency. Upon finding such inconsistency, the system sends a syslog error message and takes appropriate action if configured to do so. The following command was introduced: **bgp consistency-checker** The following command was modified: **show ip bgp vpnv4**.



CHAPTER 91

BGP—Origin AS Validation

The BGP—Origin AS Validation feature helps prevent network administrators from inadvertently advertising routes to networks they do not control. This feature uses a Resource Public Key Infrastructure (RPKI) server to authenticate that certain BGP prefixes originated from an expected autonomous system before the prefixes are allowed to be advertised.

Information About BGP Origin AS Validation

Benefit of BGP—Origin AS Validation

Occasionally network administrators have unintentionally advertised routes to networks that they do not control. This security issue can be avoided by configuring the BGP—Origin AS Validation feature. This feature uses an RPKI server to authenticate certain BGP prefixes as having originated from an expected autonomous system before prefixes are accepted.

How BGP—Origin AS Validation Works

The network administrator must set up a Resource Public Key Infrastructure (RPKI) server, using third-party software. The RPKI server handles the actual authentication of public key certificates. The server is set up so that certain prefixes or prefix ranges are allowed to originate from certain autonomous systems.

The administrator then configures the router to establish a TCP connection to the RPKI server. This is done by configuring the **bgp rpki server** command. Upon such configuration or booting the router, the router opens a TCP connection to the indicated IP address and port number. The router downloads a list of prefixes and permitted origin AS numbers from one or more router/RPKI servers using the RPKI-Router protocol (RTR). Thus, the router obtains information from the server about which autonomous systems are permitted to advertise which routes, that is, from which AS a route may originate.

If the TCP connection attempt fails, the router retries the connection once per minute. In the meantime, BGP will behave without performing origin validation.

After the TCP session between the router and the server is established, the server will normally send to the router incremental updates with new prefixes that have been added to the RPKI database. The router might also query the server every refresh interval. The router will not send a serial query message or reset query message during the interval between when it sends a serial query or reset query and when it receives an End of Data (EOD) message. Serial queries in this interval are stripped, and reset queries in this interval are sent upon receipt of the EOD message.

A prefix or prefix range and the origin-AS corresponding to it are considered an SOVC record. Overlapping prefix ranges are allowed. An SOVC table containing three records might look like this:

10.0.1.0/20-25 AS 3

10.0.1.0/19-24 AS 4

10.0.1.0/23-27 AS 5

When a prefix (network) is received from an external BGP (eBGP) peer, the prefix is initially placed in the Not Found state. It is then examined and marked as Valid, Invalid, or Not Found:

- Valid—Indicates the prefix and AS pair are found in the SOVC table.
- Invalid—Indicates the prefix meets either of the following two conditions: 1. It matches one or more Route Origin Authorizations (ROAs), but there is no matching ROA where the origin AS matches the origin AS on the AS-PATH. 2. It matches the one or more ROAs at the minimum-length specified in the ROA, but for all ROAs where it matches the minimum length, it is longer than the specified maximum length. Origin AS does not matter for condition #2.
- Not Found—Indicates the prefix is not among the valid or invalid prefixes.

By default, a prefix that is marked Invalid is not advertised to any peer, will be withdrawn from the BGP routing table if it was already advertised, and will not be flagged as a bestpath or considered as a candidate for multipath (unless a BGP bestpath command indicates otherwise). Unless a BGP bestpath command is configured indicating otherwise, the bestpath computation prefers Valid prefixes over Not Found prefixes, and both types of prefixes are advertised.

A prefix marked as Valid is installed in the BGP routing table.

By default, a prefix marked as Not Found is installed in the BGP routing table and will only be flagged as a bestpath or considered as a candidate for multipath if there is no Valid alternative (independently of other BGP attributes such as Local Preference or ASPATH).

If more than one RPKI server is configured, the router will connect to all configured servers and download prefix information from all of them. The SOVC table will be made of the union of all the records received from the different servers.

Once the **bgp rpki server** command (or the **neighbor announce rpki state** command) is configured for an address family, the router starts doing RPKI validation for every path in that address family.

Option to Announce RPKI Validation State to Neighbors

You may optionally announce (and receive) the validation state of a prefix to (and from) internal BGP (iBGP) neighbors by using an extended community attribute. This option might be more convenient for some routers than configuring the **bgp rpki server** command, because it saves that router from having to connect to an RPKI server.

The **neighbor announce rpki state** command causes the router to send the RPKI status with the route to its iBGP neighbors in the BGP extended community attribute. The router also receives RPKI status with the

route from its iBGP neighbor. The announcement works in both directions. The extended community attribute announced is:

0x4300 0x0000 (4 bytes indicating state)

The four bytes indicating state are treated as a 32-bit unsigned integer having one of the following values:

- 0—Valid
- 1—Not Found
- 2—Invalid

If the **neighbor announce rpki state** command is configured, upon receiving a route with this extended community attribute attached from an iBGP peer, the router assigns the route the corresponding validation state. If the **neighbor announce rpki state** command is not configured, all prefixes received from an iBGP peer will be marked as Valid, including the prefixes that must have marked as Not Found.

Note This extended community attribute is not sent to eBGP neighbors, even if they are configured to allow sending of this attribute.

The RPKI state extended community follows these additional behaviors:

- The configuration of the **neighbor announce rpki state** command is possible only if the router is configured to send extended communities to that neighbor on that address family.
- The **neighbor announce rpki state** command is completely independent of whether RPKI is configured for the address family.
- Once the **neighbor announce rpki state** command or the **bgp rpki server** command is configured for an address family, the router starts doing RPKI validation for every path in that address family.
- The enabling and disabling of the **neighbor announce rpki state** command causes neighbors to be split into their own update groups based on whether this portion of their configuration is identical.
- If the **neighbor announce rpki state** command is not configured, the router will save the RPKI state received from other routers, but will use it only if at least one other neighbor in the address family is configured with the **neighbor announce rpki state** command or if the topology is otherwise enabled for the use of RPKI.
- If the **neighbor send-community extended** or **neighbor send-community both** command is removed from the configuration, the **neighbor announce rpki state** configuration is also removed.
- When configuring a route reflector (RR), if the RR server receives a network that includes an RPKI state extended community from a client for whom the **neighbor announce rpki state** command is not configured, the RR will reflect the extended community to all its clients that are capable of receiving it.
- If a network has an RPKI state extended community and is received by an RR from a neighbor for which the **neighbor announce rpki state** command is configured, then it will be reflected to all RR clients that are configured to accept extended communities, regardless of whether the **neighbor announce rpki state** command is configured for those other RR clients.
- A **neighbor announce rpki state** command can be used in a peer policy template, and it is inherited.
- If a **neighbor announce rpki state** command is used in a peer policy template, it must be in the same template as the **send-community extended** command. The **neighbor announce rpki state** command

and the **send-community extended** command must come from the same template or be configured for the same neighbor.

Use of the Validation State in BGP Best Path Determination

There are two ways you can modify the default BGP best path selection process when using RPKI validation states:

- You can completely disable the validation of prefixes by the RPKI server and the storage of that validation information. This is done by configuring the **bgp bestpath prefix-validate disable** command. You might want to do this for configuration testing. The router will still connect to the RPKI server and download the validation information, but will not use the information.
- You can allow an invalid prefix to be used as the BGP best path, even if valid prefixes are available. This is the default behavior. The command to allow a BGP best path to be an invalid prefix, as determined by the BGP Origin AS Validation feature, is the **bgp bestpath prefix-validate allow-invalid** command. The prefix validation state will still be assigned to paths, and will still be communicated to iBGP neighbors that have been configured to receive RPKI state information. You can use a route map to set a local preference, metric, or other property based on the validation state.

During BGP best path selection, the default behavior, if neither of the above options is configured, is that the system will prefer prefixes in the following order:

- Those with a validation state of valid.
- Those with a validation state of not found.
- Those with a validation state of invalid (which, by default, will not be installed in the routing table).

These preferences override metric, local preference, and other choices made during the bestpath computation. The standard bestpath decision tree applies only if the validation state of the two paths is the same.

If both commands are configured, the **bgp bestpath prefix-validate disable** command will prevent the validation state from being assigned to paths, so the **bgp bestpath prefix-validate allow-invalid** command will have no effect.

These configurations can be in either router configuration mode or in address family configuration mode for the IPv4 unicast or IPv6 unicast address families.

Use of a Route Map to Customize Treatment of Valid and Invalid Prefixes

You can create a route map to match on any of the RPKI states, and thereby create a custom policy for handling valid or invalid prefixes.

By default, the router overrides all other preferences to reject routes that are in an invalid state. You must explicitly configure the **bgp bestpath prefix-validate allow-invalid** command if you want to use a route map to do something such as permit such prefixes, but with a nondefault local preference.

How to Configure BGP Origin AS Validation

Enabling BGP—Origin AS Validation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp rpki server tcp** {*ipv4-address* | *ipv6-address*} **port** *port-number* **refresh** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp rpki server tcp** {*ipv4-address* | *ipv6-address*} **port** *port-number* **refresh** *seconds* **Example:** `Device(config-router)# bgp rpki server tcp 192.168.2.2 port 1029 refresh 600`	Configures the router to connect to the specified RPKI server and download prefix information at intervals specified by the **refresh** *seconds* keyword and argument.

Announcing the RPKI State to iBGP Neighbors

Perform this task to cause the router to announce the RPKI state with routes to its iBGP neighbors in the BGP extended community attribute and to also receive the RPKI state with routes from iBGP neighbors. This task might be more convenient than configuring the BGP—Origin AS Validation feature on the router.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address*} **send-community extended**
5. **neighbor** {*ip-address* | *ipv6-address*} **announce rpki state**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** {*ip-address* | *ipv6-address*} **send-community extended** **Example:** `Device(config-router)# neighbor 192.168.1.2 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 5	**neighbor** {*ip-address* | *ipv6-address*} **announce rpki state** **Example:** `Device(config-router)# neighbor 192.168.1.2 announce rpki state`	Causes the router to send and receive the RPKI state to and from its iBGP neighbor in the BGP extended community attribute.

Disabling the Validation of BGP Prefixes, But Still Downloading RPKI Information

Perform this task if the BGP—Origin AS Validation feature is enabled, but you want to disable the validation of prefixes based on origin AS and disable the storage of validation information. The router will still connect to the RPKI server and still download the validation information, but the information will not be used in any way. This task is useful for configuration testing.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** {**ipv4** | **ipv6**} **unicast**
5. **bgp bestpath prefix-validate disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family** {**ipv4**	**ipv6**} **unicast** **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 5	**bgp bestpath prefix-validate disable** **Example:** `Device(config-router-af)# bgp bestpath prefix-validate disable`	Disables the validation of prefixes and the storage of validation information.

Allowing Invalid Prefixes as the Best Path

Perform this task if the BGP—Origin AS Validation feature is enabled, and you want to allow invalid prefixes to be used as the best path, even if valid prefixes are available. Thus, you have control over announcing invalid networks, but preferring them less than valid and not-found prefixes. Also, the downstream peer can modify path attributes based on a route map that matches invalid prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** {**ipv4** | **ipv6**} **unicast**
5. **bgp bestpath prefix-validate allow-invalid**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family** {**ipv4** | **ipv6**} **unicast** **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 5	**bgp bestpath prefix-validate allow-invalid** **Example:** `Device(config-router-af)# bgp bestpath prefix-validate allow-invalid`	Allows invalid prefixes to be used as the best path, even if valid prefixes are available.

Configuring a Route Map Based on RPKI States

Perform this task to create a route map based on RPKI states. The route map in this particular task sets a policy for all three RPKI states based on local preference, but other **set** commands can be used to set a policy. This task does not include a command that makes use of this route map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** {**ipv4** | **ipv6**} **unicast**
5. **bgp bestpath prefix-validate allow-invalid**
6. **exit**
7. **exit**
8. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
9. **match rpki** {**not-found** | **invalid** | **valid**}
10. **set local-preference** *number*

11. **exit**
12. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
13. **match rpki** {**not-found** | **invalid** | **valid**}
14. **set local-preference** *number*
15. **exit**
16. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
17. **match rpki** {**not-found** | **invalid** | **valid**}
18. **set local-preference** *number*
19. **exit**
20. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family** {**ipv4**	**ipv6**} **unicast** **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 5	**bgp bestpath prefix-validate allow-invalid** **Example:** `Device(config-router-af)# bgp bestpath prefix-validate allow-invalid`	Allows invalid prefixes to be used as the best path, even if valid prefixes are available. • This command is necessary to allow invalid prefixes, which are part of the example route map in Step 16.
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	Exits a configuration mode to the next highest mode in the CLI mode hierarchy.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits a configuration mode to the next highest mode in the CLI mode hierarchy.
Step 8	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME-1 permit 10`	Enters route map configuration mode and creates a route map that will permit routes that are allowed by the match clauses that follow.
Step 9	**match rpki** {**not-found** | **invalid** | **valid**} **Example:** `Device(config-route-map)# match rpki valid`	Creates a match clause to permit prefixes with the specified RPKI state. • This example matches on the RPKI state of valid.
Step 10	**set local-preference** *number* **Example:** `Device(config-route-map)# set local-preference 200`	Creates a set clause to set matched prefixes to a local preference of 200.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits a configuration mode to the next highest mode in the CLI mode hierarchy.
Step 12	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME-1 permit 20`	Continues in the same route map, but a later sequence number, and enters route map configuration mode.
Step 13	**match rpki** {**not-found** | **invalid** | **valid**} **Example:** `Device(config-route-map)# match rpki not-found`	Creates a match clause to permit prefixes with the specified RPKI state. • This example matches on the RPKI state of not found.
Step 14	**set local-preference** *number* **Example:** `Device(config-route-map)# set local-preference 100`	Sets the local preference of prefixes with the RPKI state of not found to 100.
Step 15	**exit** **Example:** `Device(config-route-map)# exit`	Exits a configuration mode to the next highest mode in the CLI mode hierarchy.

	Command or Action	Purpose
Step 16	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME-1 permit 30`	Continues in the same route map, but a later sequence number, and enters route map configuration mode.
Step 17	**match rpki** {**not-found** | **invalid** | **valid**} **Example:** `Device(config-route-map)# match rpki invalid`	Creates a match clause to permit prefixes with the specified RPKI state. • This example matches on the RPKI state of invalid.
Step 18	**set local-preference** *number* **Example:** `Device(config-route-map)# set local-preference 50`	Sets the local preference of prefixes with the RPKI state of invalid to 50.
Step 19	**exit** **Example:** `Device(config-route-map)# exit`	Exits a configuration mode to the next highest mode in the CLI mode hierarchy.
Step 20	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map ROUTE-MAP-NAME-1 permit 40`	Continues in the same route map, but a later sequence number, and enters route map configuration mode. • This example permits other routes rather than deny all other routes.
Step 21	**end** **Example:** `Device(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.

Configuration Examples for BGP Origin AS Validation

Example: Configuring BGP to Validate Prefixes Based on Origin AS

In the following example, the router is configured to connect to two RPKI servers, from which it will receive SOVC records of BGP prefixes and AS numbers.

```
router bgp 65000
 no bgp log-neighbor changes
 bgp rpki server tcp 10.0.96.254 port 32001 refresh 600
 bgp rpki server tcp FEC0::1002 port 32002 refresh 600
```

Example: Announcing RPKI State to Neighbors

```
router bgp 65000
 neighbor 10.10.10.10 remote-as 65000
 address-family ipv4 unicast
 neighbor 10.10.10.10 send-community extended
 neighbor 10.10.10.10 announce rpki state
```

Example: Disabling the Checking of Prefixes

The following example, for the IPv4 address family, disables the checking of prefixes to ensure they are valid. It also disables the storage of validation information. However, the router will still connect to the RPKI server and download the validation information. This example is useful for configuration testing.

```
router bgp 65000
 bgp rpki server tcp 10.0.96.254 port 32001 refresh 600
 address-family ipv4 unicast
 bgp bestpath prefix-validate disable
```

Example: Allowing Invalid Prefixes as Best Path

In the following example, for the IPv6 address family, invalid prefixes are allowed to be used as the best path, even if valid prefixes are available.

```
 router bgp 65000
  bgp rpki server tcp FEC0::1002 port 32002 refresh 600
  address-family ipv6 unicast
  bgp bestpath prefix-validate allow-invalid
```

Example: Using a Route Map Based on RPKI State

In the following example, a route map named rtmap-PEX1-3 sets a local preference of 50 for invalid prefix/AS pairs, 100 for not-found prefix/AS pairs, and 200 for valid prefix/AS pairs. The local preference values are set for incoming routes from the neighbor at 10.0.102.1. The neighbor at 10.0.102.1 is an eBGP peer. Note that the **bgp bestpath prefix-validate allow-invalid** command is required in order to permit invalid prefixes.

```
router bgp 65000
 address-family ipv4 unicast
 neighbor 10.0.102.1 route-map rtmap-PEX1-3 in
 bgp bestpath prefix-validate allow-invalid
!
route-map rtmap-PEX1-3 permit 10
 match rpki invalid
 set local-preference 50
!
route-map rtmap-PEX1-3 permit 20
```

```
 match rpki not-found
 set local-preference 100
!
route-map rtmap-PEX1-3 permit 30
 match rpki valid
 set local-preference 200
!
route-map rtmap-PEX1-3 permit 40
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

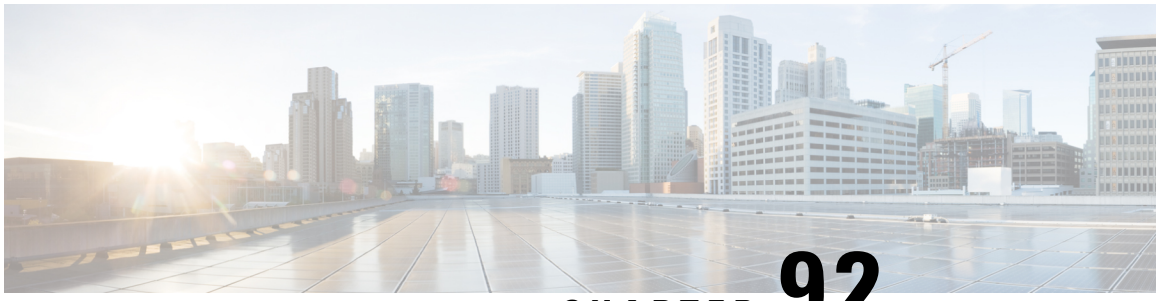
Feature Information for eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 117: Feature Information for eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

Feature Name	Releases	Feature Information
eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)		The eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6) feature allows you to configure multipath load sharing among native IPv4 and IPv6 external Border Gateway Protocol (eBGP) and internal BGP (iBGP) paths for improved load balancing in deployments. The following command was modified: **maximum-paths eibgp**.



CHAPTER 92

BGP MIB Support

The BGP MIB Support Enhancements feature introduces support in the CISCO-BGP4-MIB for new SNMP notifications.

Information About BGP MIB Support

BGP MIB Support

The Management Information Base (MIB) that supports BGP is the CISCO-BGP4-MIB. The BGP MIB Support Enhancements feature introduces support in the CISCO-BGP4-MIB for new SNMP notifications. The following sections describe the objects and notifications (traps) that are supported:

BGP FSM Transition Change Support

The cbgpRouteTable supports BGP Finite State Machine (FSM) transition state changes.

The cbgpFsmStateChange object allows you to configure SNMP notifications (traps) for all FSM transition state changes. This notification contains the following MIB objects:

- bgpPeerLastError
- bgpPeerState
- cbgpPeerLastErrorTxt
- cbgpPeerPrevState

The cbgpBackwardTransition object supports all BGP FSM transition state changes. This object is sent each time the FSM moves to either a higher or lower numbered state. This notification contains the following MIB objects:

- bgpPeerLastError

- bgpPeerState
- cbgpPeerLastErrorTxt
- cbgpPeerPrevState

The **snmp-server enable bgp traps** command allows you to enable the traps individually or together with the existing FSM backward transition and established state traps as defined in RFC 1657.

BGP Route Received Route Support

The cbgpRouteTable object supports the total number of routes received by a BGP neighbor. The following MIB object is used to query the CISCO-BGP4-MIB for routes that are learned from individual BGP peers:

- cbgpPeerAddrFamilyPrefixTable

Routes are indexed by the address-family identifier (AFI) or subaddress-family identifier (SAFI). The prefix information displayed in this table can also viewed in the output of the **show ip bgp** command.

BGP Prefix Threshold Notification Support

The cbgpPrefixMaxThresholdExceed and cbgpPrfefixMaxThresholdClear objects were introduced to allow you to poll for the total number of routes received by a BGP peer.

The cbgpPrefixMaxThresholdExceed object allows you to configure SNMP notifications to be sent when the prefix count for a BGP session has exceeded the configured value. This notification is configured on a per address family basis. The prefix threshold is configured with the **neighbor maximum-prefix** command. This notification contains the following MIB objects:

- cbgpPeerPrefixAdminLimit
- cbgpPeerPrefixThreshold

The cbgpPrfefixMaxThresholdClear object allows you to configure SNMP notifications to be sent when the prefix count drops below the clear trap limit. This notification is configured on a per address family basis. This notification contains the following objects:

- cbgpPeerPrefixAdminLimit
- cbgpPeerPrefixClearThreshold

Notifications are sent when the prefix count drops below the clear trap limit for an address family under a BGP session after the cbgpPrefixMaxThresholdExceed notification is generated. The clear trap limit is calculated by subtracting 5 percent from the maximum prefix limit value configured with the **neighbor maximum-prefix** command. This notification will not be generated if the session goes down for any other reason after the cbgpPrefixMaxThresholdExceed is generated.

VPNv4 Unicast Address Family Route Support

The cbgpRouteTable object allows you to configure SNMP GET operations for VPNv4 unicast address-family routes.

The following MIB object allows you to query for multiple BGP capabilities (for example, route refresh, multiprotocol BGP extensions, and graceful restart):

- cbgpPeerCapsTable

The following MIB object allows you to query for IPv4 and VPNv4 address family routes:

- cbgpPeerAddrFamilyTable

Each route is indexed by peer address, prefix, and prefix length. This object indexes BGP routes by the AFI and then by the SAFI. The AFI table is the primary index, and the SAFI table is the secondary index. Each BGP speaker maintains a local Routing Information Base (RIB) for each supported AFI and SAFI combination.

cbgpPeerTable Support

The cbgpPeerTable has been modified to support the enhancements described in this document. The following new table objects are supported in the CISCO-BGP-MIB.my:

- cbgpPeerLastErrorTxt
- cbgpPeerPrevState

The following table objects are not supported. The status of these objects is listed as deprecated, and these objects are not operational:

- cbgpPeerPrefixAccepted
- cbgpPeerPrefixDenied
- cbgpPeerPrefixLimit
- cbgpPeerPrefixAdvertised
- cbgpPeerPrefixSuppressed
- cbgpPeerPrefixWithdrawn

How to Enable BGP MIB Support

Enabling BGP MIB Support

SNMP notifications can be configured on the router and GET operations can be performed from an external management station only after BGP SNMP support is enabled. Perform this task on a router to configure SNMP notifications for the BGP MIB.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps bgp** [[**state-changes** [**all**] [**backward-trans**] [**limited**]] | [**threshold prefix**]]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps bgp** [[**state-changes** [**all**] [**backward-trans**] [**limited**]] | [**threshold prefix**]] **Example:** `Device(config)# snmp-server enable traps bgp`	Enables BGP support for SNMP operations. Entering this command with no keywords or arguments enables support for all BGP events. • The **state-changes** keyword is used to enable support for FSM transition events. • The **all** keyword enables support for FSM transitions events. • The **backward-trans** keyword enables support only for backward transition state change events. • The **limited** keyword enables support for backward transition state changes and established state events. • The **threshold** and **prefix** keywords are used to enable notifications when the configured maximum prefix limit is reached on the specified peer.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode, and enters privileged EXEC mode.

Configuration Examples for BGP MIB Support

Example: Enabling BGP MIB Support

The following example enables SNMP support for all supported BGP events:

```
Device(config)# snmp-server enable traps bgp
```

The following verification example shows that SNMP support for BGP is enabled by displaying any lines in the running configuration file that include “snmp-server”:

```
Device# show run | include snmp-server

snmp-server enable traps bgp
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	IP Routing: BGP Command Reference
MIB objects supported in CISCO-BGP-MIBv8.1	"Cisco-BGP-MIBv2" module in the *IP Routing: BGP Configuration Guide*
Information about SNMP and SNMP operations	SNMP Configuration Guide in the *Network Management Configuration Guide Library*

MIBs

MIB	MIBs Link
CISCO-BGP4-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

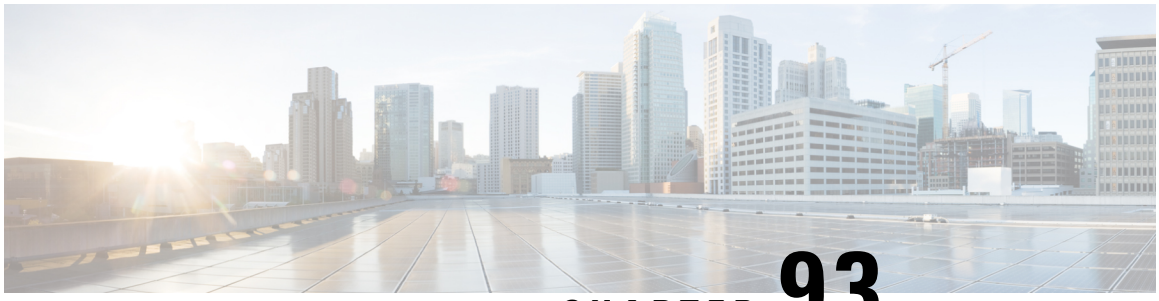
Feature Information for BGP MIB Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 118: Feature Information for BGP MIB Support

Feature Name	Releases	Feature Information
BGP MIB Support Enhancements	12.0(26)S 12.2(25)S 12.3(7)T 12.2(33)SRA 12.2(22)SXH 15.0(1)SY	The BGP MIB Support Enhancements feature introduced support in the CISCO-BGP4-MIB for new SNMP notifications. The following command was introduced: **snmp-server enable traps bgp**.



CHAPTER 93

BGP 4 MIB Support for Per-Peer Received Routes

This document describes BGP 4 MIB support for per-peer received routes. This feature introduces a table in the CISCO-BGP4-MIB that provides the capability to query (by using Simple Network Management Protocol [SNMP] commands) for routes that are learned from individual Border Gateway Protocol (BGP) peers.

Restrictions on BGP 4 MIB Support for Per-Peer Received Routes

BGP 4 MIB Support for per-Peer Received Routes supports only routes that are contained in IPv4 AFIs and unicast SAFIs in the local BGP RIB table. The BGP 4 MIB Support for per-Peer Received Routes enhancement is supported only by BGP Version 4.

Information About BGP 4 MIB Support for Per-Peer Received Routes

Overview of BGP 4 MIB Support for Per-Peer Received Routes

The BGP 4 MIB support for per-peer received routes feature introduces a table in the CISCO-BGP4-MIB that provides the capability to query (by using SNMP commands) for routes that are learned from individual BGP peers.

Before this new MIB table was introduced, a network operator could obtain the routes learned by a local BGP-speaking router by querying the local BGP speaker with an SNMP command (for example, the **snmpwalk** command). The network operator used the SNMP command to query the bgp4PathAttrTable of the CISCO-BGP4-MIB. The routes that were returned from a bgp4PathAttrTable query were indexed in the following order:

- Prefix
- Prefix length

- Peer address

Because the bgp4PathAttrTable indexes the prefixes first, obtaining routes learned from individual BGP peers will require the network operator to "walk through" the complete bgp4PathAttrTable and filter out routes from the interested peer. A BGP Routing Information Base (RIB) could contain 10,000 or more routes, which makes a manual "walk" operation impossible and automated walk operations very inefficient.

BGP 4 MIB Support for per-Peer Received Routes introduces a Cisco-specific enterprise extension to the CISCO-BGP4-MIB that defines a new table called the cbgpRouterTable. The cbgpRouterTable provides the same information as the bgp4PathAttrTable with the following two differences:

- Routes are indexed in the following order:
  - Peer address
  - Prefix
  - Prefix length

The search criteria for SNMP queries of local routes are improved because peer addresses are indexed before prefixes. A search for routes that are learned from individual peers is improved with this enhancement because peer addresses are indexed before prefixes. A network operator will no longer need to search through potentially thousands of routes to obtain the learned routes of a local BGP RIB table.

- Support is added for multiprotocol BGP, Address Family Identifier (AFI), and Subsequent Address Family Identifier (SAFI) information. This information is added in the form of indexes to the cbgpRouterTable. The CISCO-BGP4-MIB can be queried for any combination of AFIs and SAFIs that are supported by the local BGP speaker.

Note The MIB will be populated only if the router is configured to run a BGP process. The present implementation of BGP 4 MIB Support for Per-Peer Received Routes will show only routes contained in IPv4 AFI and unicast SAFI BGP local RIB tables. Support for showing routes contained in other local RIB tables will be added in the future.

BGP 4 Per-Peer Received Routes Table Elements and Objects

The following sections describe new table elements, AFI and SAFI tables and objects, and network address prefixes in the Network Layer Reachability Information (NLRI) fields that have been introduced by the BGP 4 MIB Support for Per-Peer Received Routes enhancement.

MIB Tables and Objects

The table below describes the MIB indexes of the cbgpRouterTable.

For a complete description of the MIB, see the CISCO-BGP4-MIB file CISCO-BGP4-MIB.my, available through Cisco.com at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Table 119: MIB Indexes of the cbgpRouterTable

MIB Indexes	Description
cbgpRouteAfi	Represents the AFI of the network layer protocol that is associated with the route.
cbgpRouteSafi	Represents the SAFI of the route. It gives additional information about the type of the route. The AFI and SAFI are used together to determine which local RIB (Loc-RIB) contains a particular route.
cbgpRoutePeerType	Represents the type of network layer address that is stored in the cbgpRoutePeer object.
cbgpRoutePeer	Represents the network layer address of the peer from which the route information has been learned.
cbgpRouteAddrPrefix	Represents the network address prefix that is carried in a BGP update message. See the table below for information about the types of network layer addresses that can be stored in specific types of AFI and SAFI objects.
cbgpRouteAddrPrefixLen	Represents the length in bits of the network address prefix in the NLRI field. See the table below for a description of the 13 possible entries.

AFIs and SAFIs

The table below lists the AFI and SAFI values that can be assigned to or held by the cbgpRouteAfi and cbgpRouteSafi indexes, respectively. The table below also displays the network address prefix type that can be held by specific combinations of AFIs and SAFIs. The type of network address prefix that can be carried in a BGP update message depends on the combination of AFIs and SAFIs.

Table 120: AFIs and SAFIs

AFI	SAFI	Type
ipv4(1)	unicast(1)	IPv4 address
ipv4(1)	multicast(2)	IPv4 address
ipv4(1)	vpn(128)	VPN-IPv4 address
ipv6(2)	unicast(1)	IPv6 address

Note A VPN-IPv4 address is a 12-byte quantity that begins with an 8-byte Route Distinguisher (RD) and ends with a 4-byte IPv4 address. Any bits beyond the length specified by cbgpRouteAddrPrefixLen are represented as zeros.

Network Address Prefix Descriptions for the NLRI Field

The table below describes the length in bits of the network address prefix in the NLRI field of the cbgpRouteTable. Each entry in the table provides information about the route that is selected by any of the six indexes in the table below.

Table 121: Network Address Prefix Descriptions for the NLRI Field

Table or Object (or Index)	Description
cbgpRouteOrigin	The ultimate origin of the route information.
cbgpRouteASPathSegment	The sequence of autonomous system path segments.
cbgpRouteNextHop	The network layer address of the autonomous system border router that traffic should pass through to get to the destination network.
cbgpRouteMedPresent	Indicates that the MULTI_EXIT_DISC attribute for the route is either present or absent.
cbgpRouteMultiExitDisc	Metric that is used to discriminate between multiple exit points to an adjacent autonomous system. The value of this object is irrelevant if the value of the cbgpRouteMedPresent object is "false(2)."
cbgpRouteLocalPrefPresent	Indicates that the LOCAL_PREF attribute for the route is either present or absent.
cbgpRouteLocalPref	Determines the degree of preference for an advertised route by an originating BGP speaker. The value of this object is irrelevant if the value of the cbgRouteLocalPrefPresent object is "false(2)."
cbgpRouteAtomicAggregate	Determines if the system has selected a less specific route without selecting a more specific route.
cbgpRouteAggregatorAS	The autonomous system number of the last BGP speaker that performed route aggregation. A value of 0 indicates the absence of this attribute.
cbgpRouteAggregatorAddrType	Represents the type of network layer address that is stored in the cbgpRouteAggregatorAddr object.
cbgpRouteAggregatorAddr	The network layer address of the last BGP 4 speaker that performed route aggregation. A value of all zeros indicates the absence of this attribute.
cbgpRouteBest	An indication of whether this route was chosen as the best BGP 4 route.
cbgpRouteUnknownAttr	One or more path attributes not understood by the local BGP speaker. A size of 0 indicates that this attribute is absent.

Benefits of BGP 4 MIB Support for Per-Peer Received Routes

- Improved SNMP Query Capabilities--The search criteria for SNMP queries for routes that are advertised by individual peers are improved because the peer address is indexed before the prefix. A network operator

will no longer need to search through potentially thousands of routes to obtain the learned routes of a local BGP RIB table.

- Improved AFI and SAFI Support--Support is added for multiprotocol BGP. AFI and SAFI are added as indexes to the table. The CISCO-BGP4-MIB can be queried for any combination of AFIs and SAFIs that are supported by the local BGP speaker.

Additional References for BGP Restart Neighbor Session After Max-Prefix Limit Reached

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2918	*Route Refresh Capability for BGP-4*
RFC 4486	*Subcodes for BGP Cease Notification Message*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP 4 MIB Support for Per-Peer Received Routes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 122: Feature Information for BGP 4 MIB Support for Per-Peer Received Routes

Feature Name	Releases	Feature Information
BGP 4 MIB support for per-peer received routes	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.
BGP received routes MIB	Cisco IOS XE Release 2.1	This feature was introduced on the Cisco ASR 1000 Series Routers.

Glossary

AFI--Address Family Identifier. Carries the identity of the network layer protocol that is associated with the network address.

BGP--Border Gateway Protocol. An interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined by RFC 1163, A Border Gateway Protocol (BGP). The current implementation of BGP is BGP Version 4 (BGP4). BGP4 is the predominant interdomain routing protocol that is used on the Internet. It supports CIDR and uses route aggregation mechanisms to reduce the size of routing tables.

MBGP--multiprotocol BGP. An enhanced version of BGP that carries routing information for multiple network layer protocols and IP multicast routes. It is defined in RFC 2858, Multiprotocol Extensions for BGP-4.

MIB--Management Information Base. A group of managed objects that are contained within a virtual information store or database. MIB objects are stored so that values can be assigned to object identifiers and to assist managed agents by defining which MIB objects should be implemented. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

NLRI--Network Layer Reachability Information. Carries route attributes that describe a route and how to connect to a destination. This information is carried in BGP update messages. A BGP update message can carry one or more NLRI prefixes.

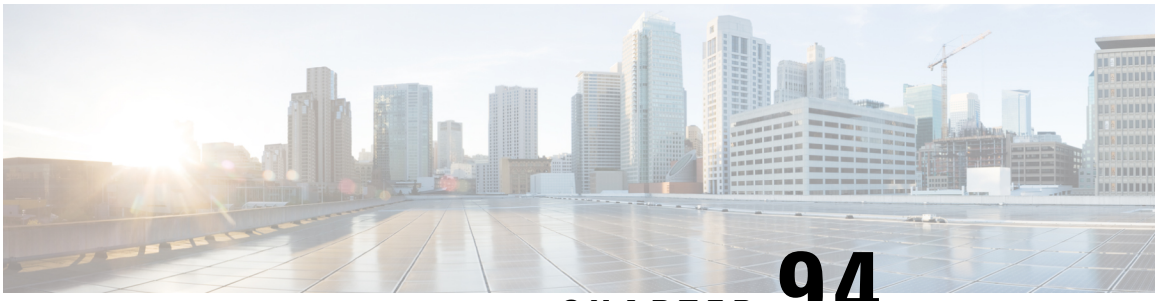
RIB--Routing Information Base (RIB). A central repository of routes that contains Layer 3 reachability information and destination IP addresses or prefixes. The RIB is also known as the routing table.

SAFI--Subsequent Address Family Identifier. Provides additional information about the type of the Network Layer Reachability Information that is carried in the attribute.

SNMP--Simple Network Management Protocol. A network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices and to manage configurations, statistics collection, performance, and security.

snmpwalk --The **snmpwalk** command is an SNMP application that is used to communicate with a network entity MIB using SNMP.

VPN--Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses a tunnel to encrypt all information at the IP level.



CHAPTER 94

BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) Using L2VPN VPLS

The BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) feature enables using L2VPN VPLS provider edge (PE) routers to maintain Border Gateway Protocol (BGP) state with customer edge (CE) routers and ensure continuous packet forwarding during a Route Processor (RP) switchover or during a planned In-Service Software Upgrade (ISSU) for a PE router. CE routers do not need to be Nonstop Forwarding (NSF)-capable or NSF-aware to benefit from BGP NSR capabilities on PE routers. Only PE routers need to be upgraded to support BGP NSR--no CE router upgrades are required. BGP NSR with SSO, thus, enables service providers to provide the benefits NSF with the additional benefits of NSR without requiring CE routers to be upgraded to support BGP graceful restart.

Prerequisites for BGP Support for NSR with SSO

- Your network must be configured to run BGP.
- Multiprotocol Layer Switching (MPLS) Layer 3 VPNs must be configured.
- You must be familiar with NSF and SSO concepts and tasks.

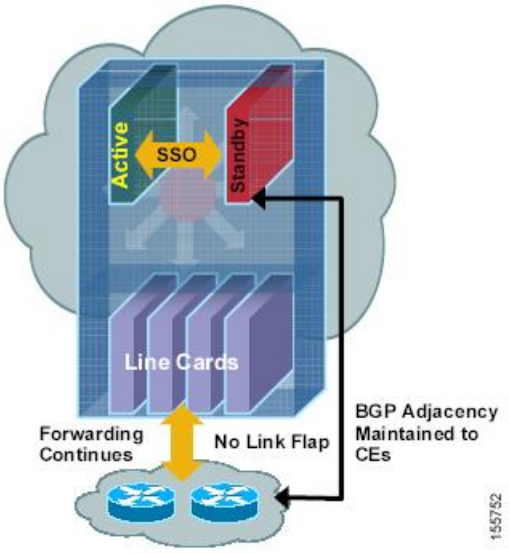
Information About BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO)

Overview of BGP NSR with SSO

Prior to the introduction of BGP NSR with SSO in Cisco IOS Release 12.2(28)SB, BGP required that all neighboring devices participating in BGP NSF be configured to be either NSF-capable or NSF-aware (by configuring the devices to support the BGP graceful restart mechanism). BGP NSF, thus, required that all neighboring devices be upgraded to a version of Cisco IOS software that supports BGP graceful restart. However, in many MPLS VPN deployments, there are situations where PE routers engage in exterior BGP (eBGP) peering sessions with CE routers that do not support BGP graceful restart and cannot be upgraded to a software version that supports BGP graceful restart in the same time frame as the provider (P) routers.

BGP NSR with SSO provides a high availability (HA) solution to service providers whose PE routers engage in eBGP peering relationships with CE routers that do not support BGP graceful restart. BGP NSR works with SSO to synchronize BGP state information between the active and standby RP. SSO minimizes the amount of time a network is unavailable to its users following a switchover. When the BGP NSR with SSO feature is configured, in the event of an RP switchover, the PE router uses BGP NSR with SSO to maintain BGP state for eBGP peering sessions with CEs that are not NSF-aware (see the figure below). Additionally, the BGP NSR with SSO feature dynamically detects NSF-aware peers and runs graceful restart with those CE routers. For eBGP peering sessions with NSF-aware peers and for internal BGP (iBGP) sessions with BGP Route Reflectors (RRs) in the service provider core, the PE uses NSF to maintain BGP state. BGP NSR with SSO, thus, enables service providers to provide the benefits of NSF with the additional benefits of NSR without requiring CE routers to be upgraded to support BGP graceful restart.

Figure 113: BGP NSR with SSO Operations During an RP Switchover



BGP NSR with SSO is supported in BGP peer, BGP peer group, and BGP session template configurations. To configure support for BGP NSR with SSO in BGP peer and BGP peer group configurations, use the **neighbor ha-mode sso** command in address family configuration mode for IPv4 VRF address family BGP

peer sessions. To include support for Cisco BGP NSR with SSO in a peer session template, use the **ha-mode sso** command in session-template configuration mode.

Benefits of BGP NSR with SSO

- Minimizes services disruptions--Border Gateway Protocol (BGP) Nonstop Routing (NSR) with Stateful Switchover (SSO) reduces impact on customer traffic during route processor (RP) switchovers (scheduled or unscheduled events), extending high availability (HA) deployments and benefits at the edge.
- Enhances high-availability Nonstop Forwarding (NSF) and SSO deployment at the edge--BGP NSR with SSO allows incremental deployment by upgrading the provider edge device with the NSR capability so that customer-facing edge devices are synchronized automatically and no coordination or NSF awareness is needed with the customer side Cisco or third-party customer edge devices. The BGP NSR feature dynamically detects NSF-aware peers and runs graceful restart with those CE devices.
- Provides transparent route convergence--BGP NSR with SSO eliminates route flaps by keeping BGP state on both active and standby RPs and ensures continuous packet forwarding with minimal packet loss during RP failovers.

How to Configure BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO)

Configuring a PE Device to Support BGP NSR with SSO

Perform this task to enable a provider edge (PE) device to maintain BGP state with customer edge (CE) devices and ensure continuous packet forwarding during a route processor (RP) switchover or during a planned ISSU. Border Gateway Protocol (BGP) Nonstop Routing (NSR) with Stateful Switchover (SSO) enables service providers to provide the benefits Nonstop Forwarding (NSF) with the additional benefits of NSR without requiring CE devices to be upgraded to support BGP graceful restart.

BGP NSR with SSO is supported in BGP peer, BGP peer group, and BGP session template configurations. Perform one of the following tasks in this section on a PE device, depending on whether you want to configure support for BGP NSR with SSO in a peer, a peer group, or a session template configuration:

Note The combination of BGP Nonstop Routing (NSR) and Stateful Switchover (SSO) is not supported. You cannot simultaneously configure the **bgp ha-mode sso** and **bgp additional-paths** [**send** | **receive**] under any address-family.

Prerequisites

- These tasks assume that you are familiar with BGP peer, BGP peer group, and BGP session template concepts. For more information, see the “Configuring a Basic BGP Network” module.
- The active and standby RP must be in SSO mode. For information about configuring SSO mode, see the “Configuring Stateful Switchover” module in the *High Availability Configuration Guide*.

- Graceful restart should be enabled on the PE device. We recommend that you enable graceful restart on all BGP peers in the provider core that participate in BGP NSF. For more information about configuring graceful restart, see the "Configuring Advanced BGP Features" module.
- CE devices must support the route refresh capability. For more information, see the "Configuring a Basic BGP Network" module.

Configuring a Peer to Support BGP NSR with SSO

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*]
5. **address-family ipv4 vrf** *vrf-name*
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **ha-mode sso**
8. **neighbor** *ip-address* **activate**
9. **end**
10. **show ip bgp vpnv4 all sso summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*] **Example:** `Device(config-router)# bgp graceful-restart`	Enables the Border Gateway Protocol (BGP) graceful restart capability and BGP Nonstop Forwarding (NSF) awareness. • If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor.

	Command or Action	Purpose
		• Use this command on the restarting device and all of its peers (NSF-capable and NSF-aware).
Step 5	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf test`	Enters address family configuration mode for IPv4 VRF address family sessions. • The **vrf**keyword and *vrf-name* argument specify that *IPv4 VRF instance information will be exchanged.* **Note** Only the syntax necessary for this task is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 7	**neighbor** *ip-address* **ha-mode sso** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 ha-mode sso`	Configures the neighbor to support BGP Nonstop Routing (NSR) with Stateful Switchover (SSO).
Step 8	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor testgroup activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local router. **Note** By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes.
Step 9	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 10	**show ip bgp vpnv4 all sso summary** **Example:** `Device# show ip bgp vpnv4 all sso summary`	(Optional) Displays the number of BGP neighbors that are in SSO mode.

Configuring a Peer Group to Support BGP NSR with SSO

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*]
5. **address-family ipv4 vrf** *vrf-name*
6. **neighbor** *peer-group-name* **peer-group**
7. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
8. **neighbor** *ip-address* **peer-group** *peer-group-name*
9. **neighbor** *peer-group-name* **ha-mode sso**
10. **neighbor** *peer-group-name* **activate**
11. **end**
12. **show ip bgp vpnv4 all sso summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*] **Example:** `Device(config-router)# bgp graceful-restart`	Enables the Border Gateway Protocol (BGP) graceful restart capability and BGP Nonstop Forwarding (NSF) awareness. • If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. • Use this command on the restarting device and all of its peers (NSF-capable and NSF-aware).
Step 5	**address-family ipv4 vrf** *vrf-name* **Example:**	Specifies the IPv4 address family and enters address family configuration mode.

	Command or Action	Purpose
	`Device(config-router)# address-family ipv4 vrf cisco`	• The **vrf**keyword and *vrf-name* argument specify that IPv4 VRF instance information will be exchanged. **Note** Only the syntax necessary for this task is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 6	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router-af)# neighbor testgroup peer-group`	Creates a BGP peer group.
Step 7	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 8	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 peer-group testgroup`	Assigns the IP address of a BGP neighbor to a BGP peer group.
Step 9	**neighbor** *peer-group-name* **ha-mode sso** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 ha-mode sso`	Configures the BGP peer group to support BGP Nonstop Routing (NSR) with Stateful Switchover (SSO).
Step 10	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor testgroup activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local device.
Step 11	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to global configuration mode.
Step 12	**show ip bgp vpnv4 all sso summary** **Example:** `Device# show ip bgp vpnv4 all sso summary`	(Optional) Displays the number of BGP neighbors that are in SSO mode.

Configuring Support for BGP NSR with SSO in a Peer Session Template

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **ha-mode sso**
6. **exit-peer-session**
7. **end**
8. **show ip bgp template peer-session** [*session-template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 101`	Enters router configuration mode and creates a Border Gateway Protocol (BGP) routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Device(config-router)# template peer-session CORE1`	Enters session-template configuration mode and creates a peer session template.
Step 5	**ha-mode sso** **Example:** `Device(config-router-stmp)# ha-mode sso`	Configures the neighbor to support BGP Nonstop Routing (NSR) with Stateful Switchover (SSO).
Step 6	**exit-peer-session** **Example:** `Device(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.
Step 7	**end** **Example:**	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router)# end`	
Step 8	**show ip bgp template peer-session** [*session-template-name*] **Example:** `Device# show ip bgp template peer-session`	(Optional) Displays locally configured peer session templates. • The output can be filtered to display a single peer policy template with the *session-template-name* argument. This command also supports all standard output modifiers.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited by or applied to another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.

For more information about configuring peer session templates, see the "Configuring a Basic BGP Network" chapter in the *Cisco IOS IP Routing: BGP Configuration Guide*.

Verifying BGP Support for NSR with SSO

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 all sso summary**
3. **show ip bgpl2vpnvpls all neighbors**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 **show ip bgp vpnv4 all sso summary**

This command is used to display the number of Border Gateway Protocol (BGP) neighbors that are in Stateful Switchover (SSO) mode.

The following is sample output from the **show ip bgp vpnv4 all sso summary** command:

Example:

```
Device# show ip bgp vpnv4 all sso summary
   Stateful switchover support enabled for 40 neighbors
```

Step 3 **show ip bgpl2vpnvpls all neighbors**

This command displays VPN address information from the BGP table.

The following is sample output from the **show ip bgp l2vpnvpls all neighbors** command. The "Stateful switchover support" field indicates whether SSO is enabled or disabled. The "SSO Last Disable Reason" field displays information about the last BGP session that lost SSO capability.

Example:

```
Device# show ip bgp l2vpn vpls all neighbors 10.3.3.3
BGP neighbor is 10.3.3.3,  vrf vrf1,  remote AS 3, external link
  Inherits from template 10vrf-session for session parameters
   BGP version 4, remote router ID 10.1.105.12
   BGP state = Established, up for 04:21:39
   Last read 00:00:05, last write 00:00:09, hold time is 30, keepalive interval is 10 seconds
   Configured hold time is 30, keepalive interval is 10 seconds
   Minimum holdtime from neighbor is 0 seconds
   Neighbor capabilities:
     Route refresh: advertised and received(new)
     Address family IPv4 Unicast: advertised and received
     Stateful switchover support enabled
   Message statistics:
     InQ depth is 0
     OutQ depth is 0
                         Sent       Rcvd
     Opens:                 1          1
     Notifications:         0          0
     Updates:               1          4
     Keepalives:         1534       1532
     Route Refresh:         0          0
     Total:              1536       1537
   Default minimum time between advertisement runs is 30 seconds
  For address family: L2VPN VPLS
   BGP table version 25161, neighbor version 25161/0
   Output queue size : 0
   Index 7, Offset 0, Mask 0x80
   7 update-group member
   Inherits from template 10vrf-policy
   Overrides the neighbor AS with my AS before sending updates
   Outbound path policy configured
   Route map for outgoing advertisements is Deny-CE-prefixes
                                Sent       Rcvd
   Prefix activity:             ----       ----
     Prefixes Current:            10         50 (Consumes 3400 bytes)
     Prefixes Total:              10         50
     Implicit Withdraw:            0          0
     Explicit Withdraw:            0          0
     Used as bestpath:           n/a          0
     Used as multipath:          n/a          0
                                  Outbound    Inbound
   Local Policy Denied Prefixes:  --------    -------
     route-map:                        150          0
     AS_PATH loop:                     n/a        760
     Total:                            150        760
   Number of NLRIs in the update sent: max 10, min 10
   Address tracking is enabled, the RIB does have a route to 10.3.3.3
   Address tracking requires at least a /24 route to the peer
   Connections established 1; dropped 0
   Last reset never
   Transport(tcp) path-mtu-discovery is enabled
   TCP session must be opened passively
Connection state is ESTAB, I/O status: 1, unread input bytes: 0 Connection is ECN Disabled Local
host: 10.0.21.1, Local port: 179 Foreign host: 10.0.21.3, Foreign port: 51205 Connection tableid
(VRF): 1
```

```
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x1625488):
Timer          Starts    Wakeups            Next
Retrans          1746        210             0x0
TimeWait            0          0             0x0
AckHold          1535       1525             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
iss: 2241977291  snduna: 2242006573  sndnxt: 2242006573     sndwnd:  13097
irs:  821359845  rcvnxt:  821391670  rcvwnd:      14883  delrcvwnd:   1501
SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms Status Flags: passive open, retransmission timeout,
gen tcbs
   0x1000
Option Flags: VRF id set, always push, md5
Datagrams (max data segment is 4330 bytes):
Rcvd: 3165 (out of order: 0), with data: 1535, total data bytes: 31824
Sent: 3162 (retransmit: 210 fastretransmit: 0),with data: 1537, total data
bytes: 29300
SSO Last Disable Reason: Application Disable (Active)
```

Troubleshooting Tips

To troubleshoot BGP NSR with SSO, use the following commands in privileged EXEC mode, as needed:

- **debug ip bgp sso** --Displays BGP-related SSO events or debugging information for BGP-related interactions between the active RP and the standby RP. This command is useful for monitoring or troubleshooting BGP sessions on a PE router during an RP switchover or during a planned ISSU.
- **debug ip tcp ha** --Displays TCP HA events or debugging information for TCP stack interactions between the active RP and the standby RP. This is command is useful for troubleshooting SSO-aware TCP connections.
- **show tcp** --Displays the status of TCP connections. The display output will display the SSO capability flag and will indicate the reason that the SSO property failed on a TCP connection.
- **show tcp ha connections** --Displays connection-ID-to-TCP mapping data.

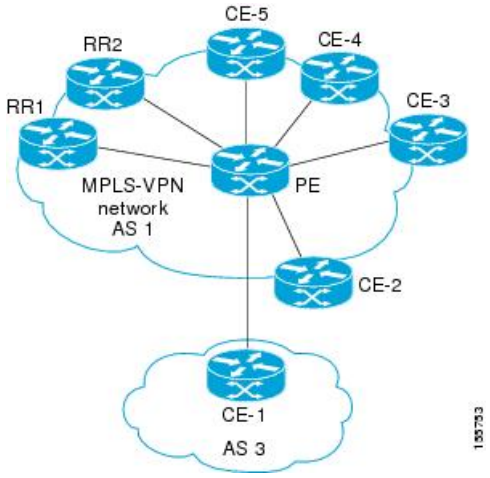
Configuration Examples for BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO)

Configuring BGP NSR with SSO Example Using L2VPN VPLS

The figure below illustrates a sample BGP NSR with SSO network topology, and the configuration examples that follow show configurations from three routers in the topology: the RR1 router, the PE router, and the CE-1 router.

Note The configuration examples omit some of the configuration required for MPLS VPNs because the purpose of these examples is to illustrate the configuration of BGP NSR with SSO.

Figure 114: BGP NSR with SSO Example Topology



RR1 Configuration

The following example shows the BGP configuration for RR1 in the figure above. RR1 is configured as a NSF-aware route reflector. In the event of an RP switchover, the PE router uses NSF to maintain the BGP state of the internal peering session with RR1.

```
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.2.2.2 remote-as 1
 neighbor 10.2.2.2 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community both
 neighbor 10.2.2.2 route-reflector-client
 exit-address-family
 !
```

PE Configuration

The following example shows the BGP NSR with SSO configuration for the PE router in the figure above. The PE router is configured to support both NSF-awareness and the BGP NSR with SSO capability. In the event of an RP switchover, the PE router uses BGP NSR with SSO to maintain BGP state for the eBGP peering session with the CE-1 router, a CE router in this topology that is not NSF-aware, and uses NSF to maintain BGP state for the iBGP session with RR1. The PE router also detects if any of the other CE routers in the MPLS VPN network are NSF-aware and runs graceful restart with those CE routers.

```
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community both
 exit-address-family
 !
 address-family l2vpn vpls
 neighbor 10.3.3.3 remote-as 3
 neighbor 10.3.3.3 ha-mode sso
 neighbor 10.3.3.3 activate
 neighbor 10.3.3.3 as-override
 no auto-summary
 no synchronization
 exit-address-family
!
```

CE-1 Configuration

The following example shows the BGP configuration for CE-1 in the figure above. The CE-1 router is configured as an external peer of the PE router. The CE-1 router is not configured to be NSF-capable or NSF-aware. The CE-1 router, however, does not need to be NSF-capable or NSF-aware to benefit from BGP NSR capabilities on the PE router nor does it need to be upgraded to support BGP NSR.

```
!
router bgp 3
 neighbor 10.2.2.2 remote-as 1
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS BGP Command Reference
MTR commands	Cisco IOS Multitopology Routing Command Reference
Configuring Multitopology Routing	Multitopology Routing Configuration Guide

Related Topic	Document Title
BGP NSR Support for iBGP Peers	BGP Configuration Guide
BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B	BGP Configuration Guide
BGP-IPV6 NSR	BGP Configuration Guide

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

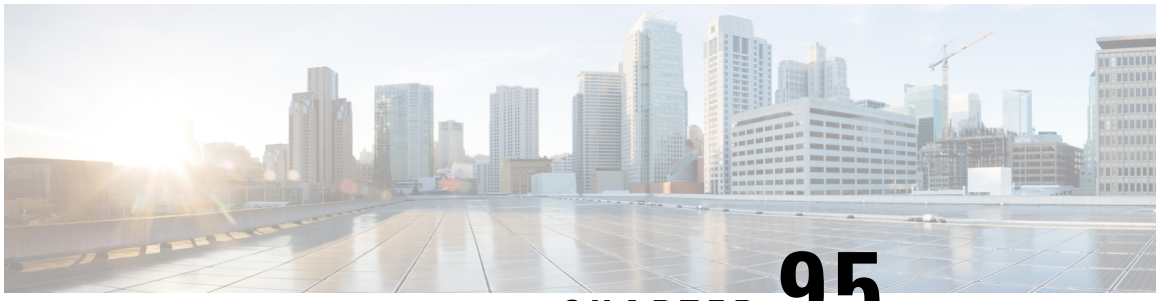
Feature Information for BGP Support for NSR with SSO

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 123: Feature Information for BGP Support for NSR with SSO

Feature Name	Releases	Feature Information
BGP Support for NSR with SSO	12.2(28)SB 15.0(1)S	The BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) feature enables provider edge (PE) routers to maintain Border Gateway Protocol (BGP) state with customer edge (CE) routers and ensure continuous packet forwarding during a Route Processor (RP) switchover or during a planned In-Service Software Upgrade (ISSU) for a PE router. CE routers do not need to be Nonstop Forwarding (NSF)-capable or NSF-aware to benefit from BGP NSR capabilities on PE routers. Only PE routers need to be upgraded to support BGP NSR--no CE router upgrades are required. BGP NSR with SSO, thus, enables service providers to provide the benefits NSF with the additional benefits of NSR without requiring CE routers to be upgraded to support BGP graceful restart. The following commands were introduced or modified: • **debug ip bgp sso** • **debug ip tcp ha** • **neighbor ha-mode sso** • **show ip bgp vpnv4** • **show ip bgp vpnv4 all sso summary** • **show tcp** • **show tcp ha connections**
BGP—NSR Enhancement	Cisco IOS Release XE 3.13S	The global support for BGP NSR and NSR preference over graceful restart has been enabled. The optional keyword **prefer** has been added to the **bgp ha-mode sso** command.



CHAPTER 95

BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) Using L2VPN VPLS

The BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) feature enables using L2VPN VPLS provider edge (PE) routers to maintain Border Gateway Protocol (BGP) state with customer edge (CE) routers and ensure continuous packet forwarding during a Route Processor (RP) switchover or during a planned In-Service Software Upgrade (ISSU) for a PE router. CE routers do not need to be Nonstop Forwarding (NSF)-capable or NSF-aware to benefit from BGP NSR capabilities on PE routers. Only PE routers need to be upgraded to support BGP NSR--no CE router upgrades are required. BGP NSR with SSO, thus, enables service providers to provide the benefits NSF with the additional benefits of NSR without requiring CE routers to be upgraded to support BGP graceful restart.

Prerequisites for BGP Support for NSR with SSO

- Your network must be configured to run BGP.
- Multiprotocol Layer Switching (MPLS) Layer 3 VPNs must be configured.
- You must be familiar with NSF and SSO concepts and tasks.

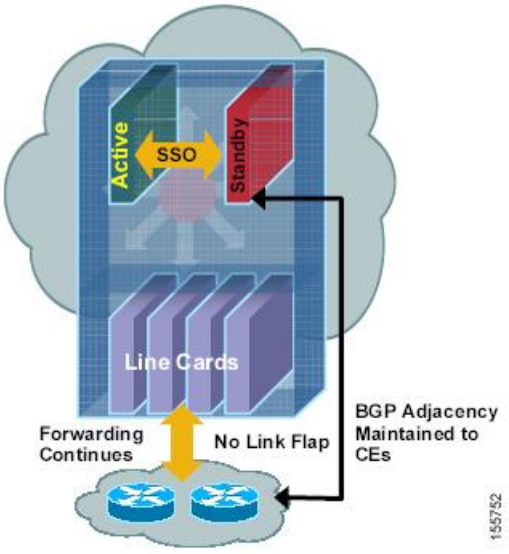
Information About BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO)

Overview of BGP NSR with SSO

Prior to the introduction of BGP NSR with SSO in Cisco IOS Release 12.2(28)SB, BGP required that all neighboring devices participating in BGP NSF be configured to be either NSF-capable or NSF-aware (by configuring the devices to support the BGP graceful restart mechanism). BGP NSF, thus, required that all neighboring devices be upgraded to a version of Cisco IOS software that supports BGP graceful restart. However, in many MPLS VPN deployments, there are situations where PE routers engage in exterior BGP (eBGP) peering sessions with CE routers that do not support BGP graceful restart and cannot be upgraded to a software version that supports BGP graceful restart in the same time frame as the provider (P) routers.

BGP NSR with SSO provides a high availability (HA) solution to service providers whose PE routers engage in eBGP peering relationships with CE routers that do not support BGP graceful restart. BGP NSR works with SSO to synchronize BGP state information between the active and standby RP. SSO minimizes the amount of time a network is unavailable to its users following a switchover. When the BGP NSR with SSO feature is configured, in the event of an RP switchover, the PE router uses BGP NSR with SSO to maintain BGP state for eBGP peering sessions with CEs that are not NSF-aware (see the figure below). Additionally, the BGP NSR with SSO feature dynamically detects NSF-aware peers and runs graceful restart with those CE routers. For eBGP peering sessions with NSF-aware peers and for internal BGP (iBGP) sessions with BGP Route Reflectors (RRs) in the service provider core, the PE uses NSF to maintain BGP state. BGP NSR with SSO, thus, enables service providers to provide the benefits of NSF with the additional benefits of NSR without requiring CE routers to be upgraded to support BGP graceful restart.

Figure 115: BGP NSR with SSO Operations During an RP Switchover



BGP NSR with SSO is supported in BGP peer, BGP peer group, and BGP session template configurations. To configure support for BGP NSR with SSO in BGP peer and BGP peer group configurations, use the **neighbor ha-mode sso** command in address family configuration mode for IPv4 VRF address family BGP

peer sessions. To include support for Cisco BGP NSR with SSO in a peer session template, use the **ha-mode sso** command in session-template configuration mode.

Benefits of BGP NSR with SSO

- Minimizes services disruptions--Border Gateway Protocol (BGP) Nonstop Routing (NSR) with Stateful Switchover (SSO) reduces impact on customer traffic during route processor (RP) switchovers (scheduled or unscheduled events), extending high availability (HA) deployments and benefits at the edge.
- Enhances high-availability Nonstop Forwarding (NSF) and SSO deployment at the edge--BGP NSR with SSO allows incremental deployment by upgrading the provider edge device with the NSR capability so that customer-facing edge devices are synchronized automatically and no coordination or NSF awareness is needed with the customer side Cisco or third-party customer edge devices. The BGP NSR feature dynamically detects NSF-aware peers and runs graceful restart with those CE devices.
- Provides transparent route convergence--BGP NSR with SSO eliminates route flaps by keeping BGP state on both active and standby RPs and ensures continuous packet forwarding with minimal packet loss during RP failovers.

How to Configure BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO)

Configuring a PE Device to Support BGP NSR with SSO

Perform this task to enable a provider edge (PE) device to maintain BGP state with customer edge (CE) devices and ensure continuous packet forwarding during a route processor (RP) switchover or during a planned ISSU. Border Gateway Protocol (BGP) Nonstop Routing (NSR) with Stateful Switchover (SSO) enables service providers to provide the benefits Nonstop Forwarding (NSF) with the additional benefits of NSR without requiring CE devices to be upgraded to support BGP graceful restart.

BGP NSR with SSO is supported in BGP peer, BGP peer group, and BGP session template configurations. Perform one of the following tasks in this section on a PE device, depending on whether you want to configure support for BGP NSR with SSO in a peer, a peer group, or a session template configuration:

Note The combination of BGP Nonstop Routing (NSR) and Stateful Switchover (SSO) is not supported. You cannot simultaneously configure the **bgp ha-mode sso** and **bgp additional-paths** [**send** | **receive**] under any address-family.

Prerequisites

- These tasks assume that you are familiar with BGP peer, BGP peer group, and BGP session template concepts. For more information, see the “Configuring a Basic BGP Network” module.
- The active and standby RP must be in SSO mode. For information about configuring SSO mode, see the “Configuring Stateful Switchover” module in the *High Availability Configuration Guide*.

- Graceful restart should be enabled on the PE device. We recommend that you enable graceful restart on all BGP peers in the provider core that participate in BGP NSF. For more information about configuring graceful restart, see the "Configuring Advanced BGP Features" module.
- CE devices must support the route refresh capability. For more information, see the "Configuring a Basic BGP Network" module.

Configuring a Peer to Support BGP NSR with SSO

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*]
5. **address-family ipv4 vrf** *vrf-name*
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **ha-mode sso**
8. **neighbor** *ip-address* **activate**
9. **end**
10. **show ip bgp vpnv4 all sso summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*] **Example:** `Device(config-router)# bgp graceful-restart`	Enables the Border Gateway Protocol (BGP) graceful restart capability and BGP Nonstop Forwarding (NSF) awareness. • If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor.

	Command or Action	Purpose
		• Use this command on the restarting device and all of its peers (NSF-capable and NSF-aware).
Step 5	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf test`	Enters address family configuration mode for IPv4 VRF address family sessions. • The **vrf**keyword and *vrf-name* argument specify that *IPv4 VRF instance information will be exchanged.* **Note** Only the syntax necessary for this task is displayed. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.
Step 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 7	**neighbor** *ip-address* **ha-mode sso** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 ha-mode sso`	Configures the neighbor to support BGP Nonstop Routing (NSR) with Stateful Switchover (SSO).
Step 8	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor testgroup activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local router. **Note** By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes.
Step 9	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 10	**show ip bgp vpnv4 all sso summary** **Example:** `Device# show ip bgp vpnv4 all sso summary`	(Optional) Displays the number of BGP neighbors that are in SSO mode.

Configuring a Peer Group to Support BGP NSR with SSO

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*]
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
7. **neighbor** *ip-address* **peer-group** *peer-group-name*
8. **neighbor** *peer-group-name* **ha-mode sso**
9. **address-family l2vpn vpls**
10. **neighbor** *peer-group-name* **activate**
11. **end**
12. **show ip bgp l2vpn vpls all sso summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**bgp graceful-restart [restart-time** *seconds*] [**stalepath-time** *seconds*] **Example:** `Device(config-router)# bgp graceful-restart`	Enables the Border Gateway Protocol (BGP) graceful restart capability and BGP Nonstop Forwarding (NSF) awareness. • If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. • Use this command on the restarting device and all of its peers (NSF-capable and NSF-aware).
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:**	Creates a BGP peer group.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor testgroup peer-group`	
Step 6	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device.
Step 7	**neighbor** *ip-address* **peer-group** *peer-group-name* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 peer-group testgroup`	Assigns the IP address of a BGP neighbor to a BGP peer group.
Step 8	**neighbor** *peer-group-name* **ha-mode sso** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 ha-mode sso`	Configures the BGP peer group to support BGP Nonstop Routing (NSR) with Stateful Switchover (SSO).
Step 9	**address-family l2vpn vpls** **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies activation of L2VPN VPLS peering.
Step 10	**neighbor** *peer-group-name* **activate** **Example:** `Device(config-router-af)# neighbor testgroup activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local device.
Step 11	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to global configuration mode.
Step 12	**show ip bgp l2vpn vpls all sso summary** **Example:** `Device# show ip bgp l2vpn vpls all sso summary`	(Optional) Displays the number of BGP neighbors that are in SSO mode.

Configuring Support for BGP NSR with SSO in a Peer Session Template

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **ha-mode sso**
6. **exit-peer-session**
7. **end**
8. **show ip bgp template peer-session** [*session-template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 101`	Enters router configuration mode and creates a Border Gateway Protocol (BGP) routing process.
Step 4	**template peer-session** *session-template-name* **Example:** `Device(config-router)# template peer-session CORE1`	Enters session-template configuration mode and creates a peer session template.
Step 5	**ha-mode sso** **Example:** `Device(config-router-stmp)# ha-mode sso`	Configures the neighbor to support BGP Nonstop Routing (NSR) with Stateful Switchover (SSO).
Step 6	**exit-peer-session** **Example:** `Device(config-router-stmp)# exit-peer-session`	Exits session-template configuration mode and returns to router configuration mode.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp template peer-session** [*session-template-name*]	(Optional) Displays locally configured peer session templates.

	Command or Action	Purpose
	Example: `Device# show ip bgp template peer-session`	• The output can be filtered to display a single peer policy template with the *session-template-name* argument. This command also supports all standard output modifiers.

What to Do Next

After the peer session template is created, the configuration of the peer session template can be inherited by or applied to another peer session template with the **inherit peer-session** or **neighbor inherit peer-session** command.

For more information about configuring peer session templates, see the "Configuring a Basic BGP Network" chapter in the *Cisco IOS IP Routing: BGP Configuration Guide*.

Verifying BGP Support for NSR with SSO

SUMMARY STEPS

1. **enable**
2. **show ip bgpl2vpnvpls all sso summary**
3. **show ip bgpl2vpnvpls all neighbors**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 **show ip bgpl2vpnvpls all sso summary**

This command is used to display the number of Border Gateway Protocol (BGP) neighbors that are in Stateful Switchover (SSO) mode.

The following is sample output from the **show ip bgp l2vpnvpls all sso summary** command:

Example:

```
Device# show ip bgp l2vpn vpls all sso summary
   Stateful switchover support enabled for 40 neighbors
```

Step 3 **show ip bgpl2vpnvpls all neighbors**

This command displays VPN address information from the BGP table.

The following is sample output from the **show ip bgp l2vpnvpls all neighbors** command. The "Stateful switchover support" field indicates whether SSO is enabled or disabled. The "SSO Last Disable Reason" field displays information about the last BGP session that lost SSO capability.

Example:

```
Device# show ip bgp l2vpn vpls all neighbors 10.3.3.3
BGP neighbor is 10.3.3.3,  vrf vrf1,  remote AS 3, external link
  Inherits from template 10vrf-session for session parameters
   BGP version 4, remote router ID 10.1.105.12
   BGP state = Established, up for 04:21:39
   Last read 00:00:05, last write 00:00:09, hold time is 30, keepalive interval is 10 seconds
   Configured hold time is 30, keepalive interval is 10 seconds
   Minimum holdtime from neighbor is 0 seconds
   Neighbor capabilities:
     Route refresh: advertised and received(new)
     Address family IPv4 Unicast: advertised and received
     Stateful switchover support enabled
   Message statistics:
     InQ depth is 0
     OutQ depth is 0
                          Sent       Rcvd
     Opens:                  1          1
     Notifications:          0          0
     Updates:                1          4
     Keepalives:          1534       1532
     Route Refresh:          0          0
     Total:               1536       1537
   Default minimum time between advertisement runs is 30 seconds
  For address family: L2VPN VPLS
   BGP table version 25161, neighbor version 25161/0
   Output queue size : 0
   Index 7, Offset 0, Mask 0x80
   7 update-group member
   Inherits from template 10vrf-policy
   Overrides the neighbor AS with my AS before sending updates
   Outbound path policy configured
   Route map for outgoing advertisements is Deny-CE-prefixes
                               Sent       Rcvd
   Prefix activity:            ----       ----
     Prefixes Current:           10         50 (Consumes 3400 bytes)
     Prefixes Total:             10         50
     Implicit Withdraw:           0          0
     Explicit Withdraw:           0          0
     Used as bestpath:          n/a          0
     Used as multipath:         n/a          0
                                 Outbound    Inbound
   Local Policy Denied Prefixes:    --------    -------
     route-map:                          150          0
     AS_PATH loop:                       n/a        760
     Total:                              150        760
   Number of NLRIs in the update sent: max 10, min 10
   Address tracking is enabled, the RIB does have a route to 10.3.3.3
   Address tracking requires at least a /24 route to the peer
   Connections established 1; dropped 0
   Last reset never
   Transport(tcp) path-mtu-discovery is enabled
   TCP session must be opened passively
Connection state is ESTAB, I/O status: 1, unread input bytes: 0 Connection is ECN Disabled Local
host: 10.0.21.1, Local port: 179 Foreign host: 10.0.21.3, Foreign port: 51205 Connection tableid
(VRF): 1
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x1625488):
Timer          Starts    Wakeups            Next
Retrans          1746        210             0x0
TimeWait            0          0             0x0
AckHold          1535       1525             0x0
SendWnd             0          0             0x0
```

```
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
iss: 2241977291  snduna: 2242006573  sndnxt: 2242006573     sndwnd:  13097
irs:  821359845  rcvnxt:  821391670  rcvwnd:      14883  delrcvwnd:   1501
SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms Status Flags: passive open, retransmission timeout,
gen tcbs
   0x1000
Option Flags: VRF id set, always push, md5
Datagrams (max data segment is 4330 bytes):
Rcvd: 3165 (out of order: 0), with data: 1535, total data bytes: 31824
Sent: 3162 (retransmit: 210 fastretransmit: 0),with data: 1537, total data
bytes: 29300
SSO Last Disable Reason: Application Disable (Active)
```

Troubleshooting Tips

To troubleshoot BGP NSR with SSO, use the following commands in privileged EXEC mode, as needed:

- **debug ip bgp sso** --Displays BGP-related SSO events or debugging information for BGP-related interactions between the active RP and the standby RP. This command is useful for monitoring or troubleshooting BGP sessions on a PE router during an RP switchover or during a planned ISSU.
- **debug ip tcp ha** --Displays TCP HA events or debugging information for TCP stack interactions between the active RP and the standby RP. This is command is useful for troubleshooting SSO-aware TCP connections.
- **show tcp** --Displays the status of TCP connections. The display output will display the SSO capability flag and will indicate the reason that the SSO property failed on a TCP connection.
- **show tcp ha connections** --Displays connection-ID-to-TCP mapping data.

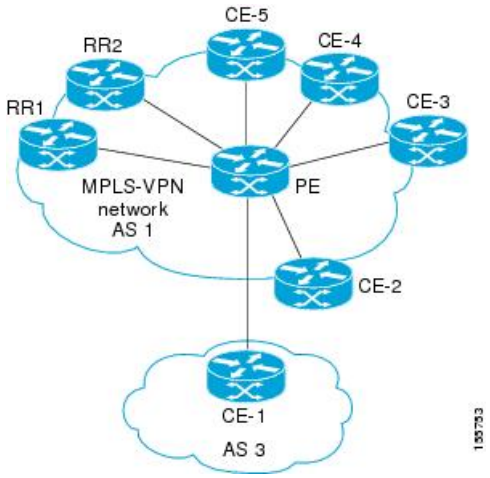
Configuration Examples for BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) using L2VPN VPLS

Example: Configuring BGP NSR with SSO Using L2VPN VPLS

The illustration below illustrates a sample Border Gateway Protocol (BGP) Nonstop Routing (NSR) with Stateful Switchful (SSO) network topology using L2VPN VPLS technology, and the configuration examples that follow show configurations from two devices in the topology: the RR1 device and the provider edge (PE) device.

Note The configuration examples omit some of the configuration required for Multiprotocol Label Switching (MPLS) VPNs because the purpose of these examples is to illustrate the configuration of BGP NSR with SSO.

Figure 116: BGP NSR with SSO Example Topology



RR1 Configuration

The following example shows the BGP configuration for RR1 in the illustration above. RR1 is configured as a Nonstop Forwarding (NSF)-aware route reflector (RR). In the event of an route processor (RP) switchover, the PE device uses NSF to maintain the BGP state of the internal peering session with RR1.

```
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.2.2.2 remote-as 1
 neighbor 10.2.2.2 update-source Loopback0
 no auto-summary
 !
 address-family l2vpn vpls
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community both
 neighbor 10.2.2.2 route-reflector-client
 exit-address-family
 !
```

PE Configuration

The following example shows the BGP NSR with SSO configuration for the PE device in the illustration above. The PE device is configured to support both NSF-awareness and the BGP NSR with SSO capability. In the event of an RP switchover, the PE device uses BGP NSR with SSO to maintain BGP state for the external BGP (eBGP) peering session and uses NSF to maintain BGP state for the internal BGP (iBGP) session with RR1.

```
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
```

```
 bgp graceful-restart
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 update-source Loopback0
 neighbor 10.3.3.3 remote-as 3
 neighbor 10.3.3.3 ha-mode sso
 neighbor 10.3.3.3 activate
 neighbor 10.3.3.3 as-override
 no auto-summary
 !
 address-family l2vpn vpls
 neighbor 10.1.1.1 activate
 neighbor 10.1.1.1 send-community both
 exit-address-family
 !
 no auto-summary
 no synchronization
 exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS BGP Command Reference
MTR commands	Cisco IOS Multitopology Routing Command Reference
Configuring Multitopology Routing	Multitopology Routing Configuration Guide
BGP NSR Support for iBGP Peers	BGP Configuration Guide
BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B	BGP Configuration Guide
BGP-IPV6 NSR	BGP Configuration Guide

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) Using L2VPN VPLS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 124: Feature Information for BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO)

Feature Name	Releases	Feature Information
BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) Using L2VPN VPLS	Cisco IOS XE Fuji 16.7.1	The BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO) using L2VPN VPLS feature enables provider edge (PE) routers to maintain Border Gateway Protocol (BGP) state with customer edge (CE) routers and ensure continuous packet forwarding during a Route Processor (RP) switchover or during a planned In-Service Software Upgrade (ISSU) for a PE router. CE routers do not need to be Nonstop Forwarding (NSF)-capable or NSF-aware to benefit from BGP NSR capabilities on PE routers. Only PE routers need to be upgraded to support BGP NSR--no CE router upgrades are required. BGP NSR with SSO, thus, enables service providers to provide the benefits NSF with the additional benefits of NSR without requiring CE routers to be upgraded to support BGP graceful restart. The following commands were modified: • **debug ip bgp sso** • **show ip bgp l2vpn**

CHAPTER 96

BGP NSR Auto Sense

The BGP NSR Auto Sense feature is the default behavior implemented to reduce unnecessary churn in the event of a Route Processor (RP) failover. Prior to this feature, when an Active RP went down, the new Active RP that was taking over to provide Border Gateway Protocol (BGP) nonstop routing (NSR) would send a route-refresh request to all peers configured with NSR. However, the new Active RP had already received all the incoming updates while acting as the Standby RP. Sending route-refresh requests caused unnecessary BGP churn during switchover; this feature prevents such route-refresh requests by default. This feature also provides NSR support to peers that lack route-refresh capability. If you want to revert to the old behavior of sending route-refresh requests, a new command is available to make that happen.

Information About BGP NSR Auto Sense

Benefits of BGP NSR Auto Sense

The BGP NSR Auto Sense feature has the following benefits:

- This feature is a default behavior that reduces unnecessary churn in the event of a Route Processor (RP) failover. Prior to this feature, when an Active RP went down, the new Active RP that was taking over to provide BGP nonstop routing (NSR) would send a route-refresh request to all peers configured with NSR. However, the Active RP had already received all the incoming updates while acting as the Standby RP. Sending route-refresh requests caused unnecessary BGP churn during switchover; this feature prevents such route-refresh requests by default.
- This feature also provides NSR support to peers that lack route-refresh capability. Prior to this feature, NSR was not supported for peers that lack route-refresh capability.
- There is no need to configure this feature; it is the default behavior in releases where this feature is implemented.
- If you want to revert to the former behavior of a new Active RP sending route-refresh requests when an RP goes down, you can use the **bgp sso route-refresh-enable** command.

Consequence of Reverting to NSR Without Auto Sense

You might have a reason not to want the default behavior of the BGP NSR Auto Sense feature. If you want to revert to the former behavior of a new Active RP sending route-refresh requests when an RP goes down, you can use the **bgp sso route-refresh-enable** command. This action causes peers that did not exchange route-refresh capability in the received OPEN message to have NSR support disabled.

How to Disable the BGP NSR Auto Sense Feature

Disabling the BGP NSR Auto Sense Feature

The BGP NSR Auto Sense feature is enabled by default. Perform this task only if you want to disable the feature, for example, if routes that were being advertised at the point of switchover did not get processed by the Standby RP (new Active RP) for some reason. In that case, sending a route-refresh to request all the routes that the peer had ever advertised would be helpful. After performing this task, in the event of a failover, a new Active RP will send route-refresh requests to peers configured with NSR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp sso route-refresh-enable**
5. **end**
6. **show ip bgp vpnv4 all neighbor** [*ip-address*]
7. **show ip bgp vpnv4 all sso summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 6500`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private

	Command or Action	Purpose
		autonomous system numbers that can be used in internal networks range from 64512 to 65535. • Regarding the 4-byte AS configuration, please see the **bgp asnotation dot** command in the *IP Routing: BGP Command Reference*.
Step 4	**bgp sso route-refresh-enable** **Example:** `Router(config-router)# bgp sso route-refresh-enable`	Disables the BGP NSR Auto Sense feature.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits to privileged EXEC mode.
Step 6	**show ip bgp vpnv4 all neighbor** [*ip-address*] **Example:** `Router# show ip bgp vpnv4 all neighbor 10.0.0.2`	(Optional) Displays information about BGP peers.
Step 7	**show ip bgp vpnv4 all sso summary** **Example:** `Router# show ip bgp vpnv4 all sso summary`	(Optional) Displays the number of BGP peers that support BGP nonstop routing (NSR) with stateful switchover (SSO).

Configuration Example for BGP NSR Auto Sense

Example: Disabling the BGP NSR Auto Sense Feature

```
router bgp 65600
 bgp sso route-refresh-enable
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

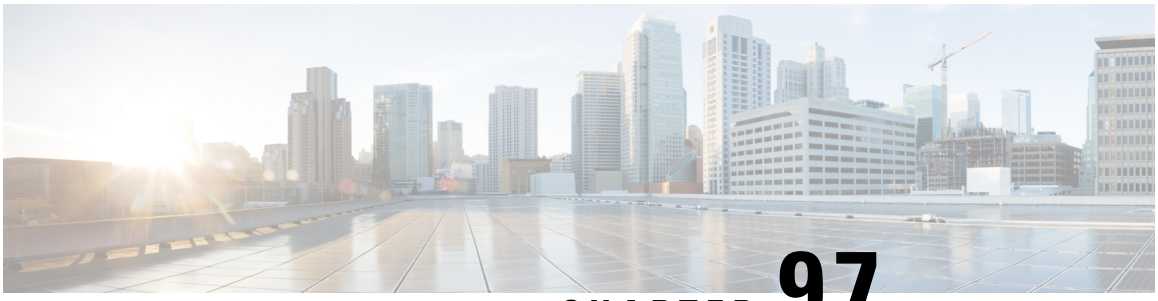
Feature Information for BGP NSR Auto Sense

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 125: Feature Information for BGP NSR Auto Sense

Feature Name	Releases	Feature Information
BGP NSR Auto Sense	15.2(2)S Cisco IOS XE Release 3.6S	The BGP NSR Auto Sense feature is implemented by default to reduce unnecessary churn in the event of an RP failover. This feature also provides NSR support to peers that lack route-refresh support. The following command was introduced: **bgp sso route-refresh-enable**.



CHAPTER 97

BGP NSR Support for iBGP Peers

BGP NSR provides BGP nonstop routing (NSR) and nonstop forwarding (NSF) in the event of a switchover from an Active RP to the Standby RP. The BGP NSR Support for iBGP Peers feature provides NSR support for iBGP peers configured under the IPv4 unicast or IPv4 + label address family.

Restrictions on BGP NSR Support for iBGP Peers

- This feature applies to iBGP peers configured under IPv4 unicast or IPv4 + label address families.
- When you configure BGP with graceful restart and remove the BGP configuration using **no router bgp** command, the graceful restart timer starts. As a result, the stale entry is present in the BGP routing table and it is only removed after the BGP graceful restart timer is over.

Information About BGP NSR Support for iBGP Peers

Benefit of BGP NSR Support for iBGP Peers

Nonstop routing is beneficial for iBGP peers because it reduces the likelihood of dropped packets during switchover from the Active RP to the Standby RP. Switchover occurs when the Active RP fails for some reason, and the Standby RP takes control of Active RP operations.

How to Configure BGP NSR Support for iBGP Peers

Making an iBGP Peer NSR-Capable for the IPv4 Address Family

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**unicast** | **vrf** *vrf-name*]
5. **neighbor** *ip-address* **remote-as** *as-number*
6. **neighbor** *ip-address* **activate**
7. **neighbor** *ip-address* **ha-mode sso**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 4000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv4** [**unicast**	**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 unicast address family. • The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:**	Specifies the autonomous system of the neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.168.1.1 remote-as 4000`	
Step 6	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Activates the specified peer.
Step 7	**neighbor** *ip-address* **ha-mode sso** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 ha-mode sso`	Configures a BGP neighbor to support BGP NSR with stateful switchover (SSO).
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Making an iBGP Peer NSR-Capable for the VPNv4 Address Family

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *as-number*
5. **neighbor** *ip-address* **ha-mode sso**
6. **address-family vpnv4** [**unicast**]
7. **neighbor** *ip-address* **activate**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 4000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.1.1 remote-as 4000`	Specifies the autonomous system of the neighbor.
Step 5	**neighbor** *ip-address* **ha-mode sso** **Example:** `Device(config-router)# neighbor 192.168.1.1 ha-mode sso`	Configures a BGP neighbor to support BGP NSR with stateful switchover (SSO).
Step 6	**address-family vpnv4** [**unicast**] **Example:** `Device(config-router)# address-family VPNv4 unicast`	Specifies the VPNv4 address family and enters address family configuration mode.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Activates the specified peer.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Making an iBGP Peer NSR Capable at the Router Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *as-number*
5. **neighbor** *ip-address* **activate**
6. **neighbor** *ip-address* **ha-mode sso**
7. **end**
8. **show ip bgp sso summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 4000`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Device(config-router)# neighbor 192.168.1.1 remote-as 4000`	Specifies the autonomous system of the neighbor.
Step 5	**neighbor** *ip-address* **activate** **Example:** `Device(config-router)# neighbor 192.168.1.1 activate`	Activates the specified neighbor.
Step 6	**neighbor** *ip-address* **ha-mode sso** **Example:** `Device(config-router)# neighbor 192.168.1.1 ha-mode sso`	Configures the specified peer to be NSR capable in all of the NSR-supported address families under which that peer has been activated.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp sso summary** **Example:** `Device# show ip bgp sso summary`	(Optional) Displays information about stateful switchover (sso) and whether a peer has NSR enabled or disabled.

Configuration Examples for BGP NSR Support for an iBGP Peer

Example: Configuring an iBGP Peer To Be NSR Capable

Configuring an iBGP Peer to Be NSR Capable at the Address Family Level

```
router bgp 4000
 address-family ipv4 unicast
 neighbor 192.168.1.1 remote-as 4000
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 ha-mode sso
```

Configuring an iBGP Peer to Be NSR Capable at the Router Level

```
router bgp 4000
 neighbor 192.168.1.1 remote-as 4000
 neighbor 192.168.1.1 activate
 neighbor 192.168.1.1 ha-mode sso
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BFD commands	Cisco IOS IP Routing: Protocol Independent Command Reference
Configuring BFD support for another routing protocol	IP Routing: BFD Configuration Guide

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

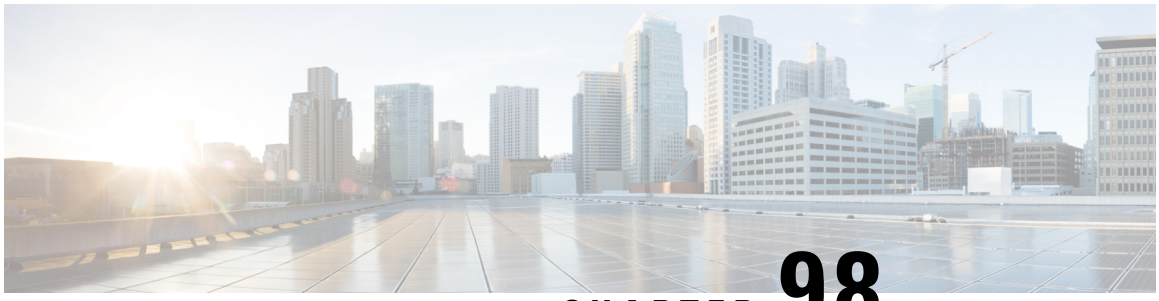
Feature Information for BGP NSR Support for iBGP Peers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 126: Feature Information for BGP NSR Support for iBGP Peers

Feature Name	Releases	Feature Information
BGP NSR Support for iBGP Peers		BGP NSR provides BGP nonstop routing and nonstop forwarding in the event of a switchover from an active RP to the standby RP. The following commands were modified: **neighbor ha-mode sso** and **show ip bgp vpnv4 all sso summary**.



CHAPTER 98

BGP Graceful Shutdown

The BGP Graceful Shutdown feature reduces or eliminates the loss of traffic along a link being shut down for maintenance. Routers always have a valid route available during the convergence process. This feature is used primarily for maintenance on a link between a Provider Edge (PE), PE-PE, PE- Route Reflector (RR), PE-Customer Edge (CE) and CE.

Information About BGP Graceful Shutdown

Purpose and Benefits of BGP Graceful Shutdown

There are times when planned maintenance operations cause routing changes in BGP. After the shutdown of eBGP and iBGP peering sessions between autonomous system border routers (ASBRs), BGP devices are temporarily unreachable during BGP convergence. The goal of gracefully shutting down one or more BGP sessions is to minimize traffic loss during the planned shutdown and subsequent reestablishment of the sessions.

The BGP Graceful Shutdown feature reduces or eliminates the loss of inbound or outbound traffic flows that were initially forwarded along the peering link that is being shut down for maintenance. This feature is primarily for PE-CE, PE-RR and PE-PE links. Lowering the local preference for paths received over the session being shutdown renders the affected paths less preferred by the BGP decision process, but still allows the paths to be used during the convergence while alternative paths are propagated to the affected devices. Therefore, devices always have a valid route available during the convergence process.

The feature also allows vendors to provide a graceful shutdown mechanism that does not require any router reconfiguration at maintenance time. The benefits of the BGP Graceful Shutdown feature are fewer lost packets and less time spent reconfiguring devices.

GSHUT Community

The GSHUT community is a well-known community used in conjunction with the BGP Graceful Shutdown feature. The GSHUT community attribute is applied to a neighbor specified by the **neighbor shutdown**

graceful command, thereby gracefully shutting down the link in an expected number of seconds. The GSHUT community is always sent by the GSHUT initiator.

The GSHUT community is specified in a community list, which is referenced by a route map and then used to make policy routing decisions.

The GSHUT community can also be used in the **show ip bgp community** command to limit output to GSHUT routes.

BGP GSHUT Enhancement

The BGP Graceful Shutdown (GSHUT) Enhancement feature enables graceful shutdown of either all neighbors or only virtual routing and forwarding (VRF) neighbors across BGP sessions. To enable the BGP GSHUT enhancement feature on the device, you must configure either the **community** keyword or the **local-preference** keyword in the **bgp graceful-shutdown all** command. Use the **activate** keyword to activate graceful shutdown either across all neighbors or only across all VRF neighbors, across all BGP sessions.

How to Configure BGP Graceful Shutdown

Shutting Down a BGP Link Gracefully

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ipv4-address* | *ipv6-address*} **remote-as** *number*
5. **neighbor** {*ipv4-address* | *ipv6-address* | *peer-group-name*} **shutdown graceful** *seconds* {**community** *value* [**local-preference** *value*] | **local-preference** *value*}
6. **end**
7. **show ip bgp community gshut**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 5000`	Configures a BGP routing process.
Step 4	**neighbor** {*ipv4-address* | *ipv6-address*} **remote-as** *number* **Example:** `Device(config-router)# neighbor 2001:db8:3::1 remote-as 5500`	Configures the autonomous system (AS) to which the neighbor belongs.
Step 5	**neighbor** {*ipv4-address* | *ipv6-address* | *peer-group-name*} **shutdown graceful** *seconds* {**community** *value* [**local-preference** *value*] | **local-preference** *value*} **Example:** `Device(config-router)# neighbor 2001:db8:3::1 shutdown graceful 600 community 1200 local-preference 300`	Configures the device to gracefully shut down the link to the specified peer in the specified number of seconds; advertises the route with the GSHUT (Graceful Shutdown) community; and advertises the route with another community or specifies a local preference value for the route, or both. • Make sure to specify an adequate amount of time for iBGP peers to converge and to choose an alternate path as the best path. • If the **graceful** keyword is used in the **neighbor shutdown** command, at least one of the two attributes (a community or local preference) must be configured. You may configure both attributes. • If the **graceful** keyword is used in the **neighbor shutdown** command, the route is advertised with the GSHUT community by default. You may also set one other community for policy routing purposes. • In this particular example, the route to the neighbor is configured to shut down in 600 seconds, is advertised with the GSHUT community and community 1200, and is configured with a local preference of 300. • The device receiving the advertisement looks at the community value(s) of the route and optionally uses the community value to apply routing policy. Filtering routes based on a community is done with the **ip community-list** command and a route map. • During the graceful shutdown, the **neighbor shutdown** command is not nvgened. After the timer expires, SHUTDOWN is nvgened.
Step 6	**end** **Example:** `Device(config-router)# end`	Returns to EXEC mode.

	Command or Action	Purpose
Step 7	**show ip bgp community gshut** **Example:** `Device# show ip bgp community gshut`	(Optional) Displays information about the routes that are advertised with the well-known GSHUT community.

Filtering BGP Routes Based on the GSHUT Community

Perform this task on a BGP peer to the device where you enabled the BGP Graceful Shutdown feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ipv4-address* | *ipv6-address*} **remote-as** *number*
5. **neighbor** {*ipv4-address* | *ipv6-address*} **activate**
6. **neighbor** {*ipv4-address* | *ipv6-address*} **send-community**
7. **exit**
8. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
9. **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]}
10. **exit**
11. **ip community-list** {*standard* | **standard** *list-name*} {**deny** | **permit**} **gshut**
12. **router bgp** *autonomous-system-number*
13. **neighbor** *address* **route-map** *map-name* **in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 2000`	Configures a BGP routing process.

	Command or Action	Purpose
Step 4	**neighbor** {*ipv4-address* | *ipv6-address*} **remote-as** *number* **Example:** `Device(config-router)# neighbor 2001:db8:4::1 remote-as 1000`	Configures the autonomous system (AS) to which the neighbor belongs.
Step 5	**neighbor** {*ipv4-address* | *ipv6-address*} **activate** **Example:** `Device(config-router)# neighbor 2001:db8:4::1 activate`	Activates the neighbor.
Step 6	**neighbor** {*ipv4-address* | *ipv6-address*} **send-community** **Example:** `Device(config-router)# neighbor 2001:db8:4::1 send-community`	Enables BGP community exchange with the neighbor.
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 8	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map RM_GSHUT deny 10`	Configures a route map to permit or deny routes for policy routing.
Step 9	**match community** {*standard-list-number* | *expanded-list-number* | *community-list-name* [**exact**]} **Example:** `Device(config-route-map)# match community GSHUT`	Configures that the routes that match ip community-list GSHUT will be policy routed.
Step 10	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode.
Step 11	**ip community-list** {*standard* | **standard** *list-name*} {**deny** | **permit**} **gshut** **Example:** `Device(config)# ip community-list standard GSHUT permit gshut`	Configures a community list and permits or denies routes that have the GSHUT community to the community list. • If you specify other communities in the same statement, there is a logical AND operation and all communities in the statement must match the communities for the route in order for the statement to be processed.

	Command or Action	Purpose
Step 12	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 2000`	Configures a BGP routing process.
Step 13	**neighbor** *address* **route-map** *map-name* **in** **Example:** `Device(config)# neighbor 2001:db8:4::1 route-map RM_GSHUT in`	Applies the route map to incoming routes from the specified neighbor. • In this example, the route map named RM_GSHUT denies routes from the specified neighbor that have the GSHUT community.

Configuring BGP GSHUT Enhancement

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp graceful-shutdown all** {**neighbors** | **vrfs**} *shutdown-time* {**community** *community-value* [**local-preference** *local-pref-value*] | **local-preference** *local-pref-value* [**community** *community-value*]}
5. **bgp graceful-shutdown all** {**neighbors** | **vrfs**} **activate**
6. **end**
7. **show ip bgp**
8. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode to create or configure a BGP routing process.

	Command or Action	Purpose
Step 4	**bgp graceful-shutdown all** {**neighbors** | **vrfs**} *shutdown-time* {**community** *community-value* [**local-preference** *local-pref-value*] | **local-preference** *local-pref-value* [**community** *community-value*]} **Example:** `Device(config-router)# bgp graceful-shutdown all neighbors 180 local-preference 20 community 10`	Enables the BGP GSHUT enhancement feature on the device.
Step 5	**bgp graceful-shutdown all** {**neighbors** | **vrfs**} **activate** **Example:** `Device(config-router)# bgp graceful-shutdown all neighbors activate`	Activates graceful shutdown across all neighbors or only across VRF neighbors for BGP sessions.
Step 6	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 7	**show ip bgp** **Example:** `Device# show ip bgp neighbors 10.2.2.2 | include shutdown`	Displays entries in the BGP routing table.
Step 8	**show running-config** **Example:** `Device# show running-config | session router bgp`	Displays running configuration on the device.

Configuration Examples for BGP Graceful Shutdown

Example: Shutting Down a BGP Link Gracefully

Graceful Shutdown While Setting a Local-Preference

This example gracefully shuts down the link to the specified neighbor in 600 seconds, adds the GSHUT community to the route, and sets a local preference of 500 for the route.

```
router bgp 1000
 neighbor 2001:db8:5::1 remote-as 2000
 neighbor 2001:db8:5::1 shutdown graceful 600 local-preference 500
 neighbor 2001:db8:5::1 send-community
 exit
```

Graceful Shutdown While Setting an Additional Community

This example gracefully shuts down the link to the specified neighbor in 600 seconds, and adds the GSHUT community and numbered community to the route.

```
router bgp 1000
 neighbor 2001:db8:5::1 remote-as 2000
 neighbor 2001:db8:5::1 shutdown graceful 600 community 1400
 neighbor 2001:db8:5::1 send-community
 exit
```

Graceful Shutdown while Setting an Additional Community and Local-Preference

This example gracefully shuts down the link to the specified neighbor in 600 seconds, adds the GSHUT community and the numbered community to the route, and sets a local preference of 500 to the route.

```
router bgp 1000
 neighbor 2001:db8:5::1 remote-as 2000
 neighbor 2001:db8:5::1 shutdown graceful 600 community 1400 local-preference 500
 neighbor 2001:db8:5::1 send-community
 exit
```

Example: Filtering BGP Routes Based on the GSHUT Community

In additional to being able to gracefully shut down a BGP route, another use of the GSHUT community is to configure a community list to filter routes with this community from getting into the BGP routing table.

This example illustrates how to use a community list to filter incoming BGP routes based on the GSHUT community. In this example, a route map named RM_GSHUT denies routes based on a standard community list named GSHUT. The community list contains routes with the GSHUT community. The route map is then applied to incoming routes from the neighbor at 2001:db8:4::1.

```
router bgp 2000
 neighbor 2001:db8:4::1 remote-as 1000
 neighbor 2001:db8:4::1 activate
 neighbor 2001:db8:4::1 send-community
 exit
route-map RM_GSHUT deny 10
 match community GSHUT
 exit
ip community-list standard GSHUT permit gshut
router bgp 2000
 neighbor 2001:db8:4::1 route-map RM_GSHUT in
```

Example: BGP GSHUT Enhancement

The following example shows how to enable and activate the BGP GSHUT enhancement feature across all neighbors. In this example, the neighbors are configured to gracefully shutdown within the specified duration of 180 seconds.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# bgp graceful-shutdown all neighbors 180 local-preference 20 community
 10
Device(config-router)# bgp graceful-shutdown all neighbors activate
Device(config-router)# end
```

Following is sample output from the **show ip bgp** command, which displays the graceful shutdown time for each neighbor. In this example, there are two IPv4 neighbors configured with IP address 10.2.2.2 and 172.16.2.1 and one VRF neighbor, tagged v1, is configured with IP address 192.168.1.1.

```
Device# show ip bgp neighbors 10.2.2.2 | include shutdown

Graceful Shutdown Timer running, schedule to reset the peer in 00:02:47 seconds
Graceful Shutdown Localpref set to 20
Graceful Shutdown Community set to 10

Device# show ip bgp neighbors 172.16.2.1 | include shutdown

Graceful Shutdown Timer running, schedule to reset the peer in 00:02:38 seconds
Graceful Shutdown Localpref set to 20
Graceful Shutdown Community set to 10

Device# show ip bgp vpnv4 vrf v1 neighbors 192.168.1.1 | include shutdown

Graceful Shutdown Timer running, schedule to reset the peer in 00:01:45 seconds
Graceful Shutdown Localpref set to 20
Graceful Shutdown Community set to 10
```

Following is sample output from the **show running-config** command, which displays information associated with the BGP session in router configuration mode:

```
Device# show running-config | session router bgp

router bgp 65000
bgp log-neighbor-changes
bgp graceful-shutdown all neighbors 180 local-preference 20 community 10
network 10.1.1.0 mask 255.255.255.0
neighbor 10.2.2.2 remote-as 40
neighbor 10.2.2.2 shutdown
neighbor 172.16.2.1 remote-as 10
neighbor 172.16.2.1 shutdown
!
address-family vpnv4
neighbor 172.16.2.1 activate
neighbor 172.16.2.1 send-community both
exit-address-family
!
address-family ipv4 vrf v1
neighbor 192.168.1.1 remote-as 30
neighbor 192.168.1.1 shutdown
```

```
neighbor 192.168.1.1 activate
neighbor 192.168.1.1 send-community both
exit-address-family
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 6198	*Requirements for the Graceful Shutdown of BGP Sessions*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

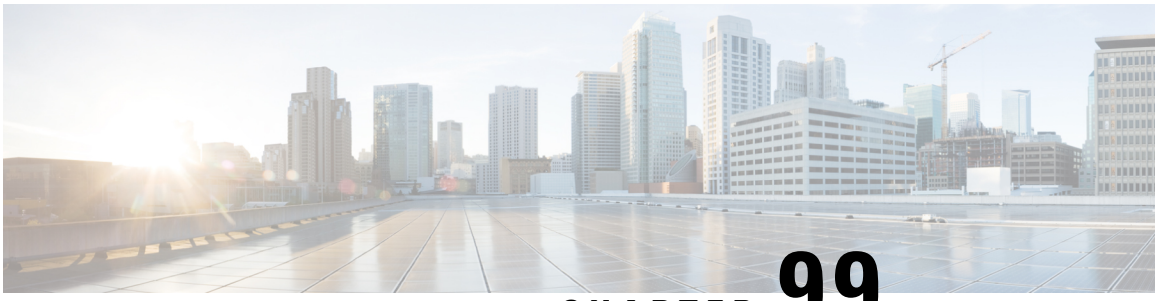
Feature Information for BGP Graceful Shutdown

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 127: Feature Information for BGP Graceful Shutdown

Feature Name	Releases	Feature Information
BGP Graceful Shutdown		The BGP Graceful Shutdown feature reduces or eliminates the loss of traffic along a link being shut down for maintenance. Routers always have a valid route available during the convergence process. The following commands were modified: **ip community-list**, **neighbor shutdown**, **show ip bgp community**, and **show ip bgp vpnv4**.
BGP GSHUT Enhancement		The BGP Graceful Shutdown (GSHUT) Enhancement feature enables graceful shutdown of either all neighbors or only virtual routing and forwarding (VRF) neighbors across BGP sessions. The following command was introduced: **bgp graceful-shutdown all**.



CHAPTER 99

BGP — mVPN BGP sAFI 129 - IPv4

The BGP—mVPN BGP sAFI 129 IPv4 feature provides the capability to support multicast routing in the service provider's core IPv4 network. This feature is needed to support BGP-based MVPNs. BGP MVPN provides a means for service providers to use different encapsulation methods (generic routing encapsulation [GRE], Multicast Label Distribution Protocol [MPDP], and ingress replication) for forwarding MVPN multicast data traffic in the service provider network.

Information About BGP--mVPN BGP sAFI 129 - IPv4

BGP — mVPN BGP sAFI 129 - IPv4 Overview

The Cisco BGP Address Family Identifier (AFI) model was introduced with multiprotocol BGP and is designed to be modular and scalable and to support multiple AFI and Subsequent Address Family Identifier (SAFI) configurations. SAFI provides additional information about the type of Network Layer Reachability Information (NLRI) that is used to describe a route and how to connect to a destination.

SAFI 129 provides the capability to support multicast routing in the service provider's core IPv4 network. This feature is needed to support BGP-based MVPNs. The addition of SAFI 129 allows multicast to select an upstream multicast hop that may be independent of the unicast topology. Multicast routes learned from the customer edge (CE) router or multicast VPN routes learned from remote provider edge (PE) routers are installed into the multicast Routing Information Base (RIB), whereas previously unicast routes in the unicast RIB were replicated into the multicast RIB.

The **address-family ipv4** command has been updated to support IP version 4 (IPv4) multicast address prefixes for a VPN routing and forwarding (VRF) instance, and the **address-family vpnv4** command has been updated to support VPN version 4 (VPNv4) multicast address prefixes.

How to Configure BGP -- mVPN BGP sAFI 129 - IPv4

Configure BGP — mVPN BGP sAFI 129 - IPv4

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf1*
4. **rd** *route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **address-family ipv4**
8. **mdt default** *group-address*
9. **exit**
10. **router bgp** *autonomous-system-number*
11. **address-family vpnv4 multicast**
12. **neighbor** *peer-group-name* **send-community extended**
13. **neighbor** *peer-group-name* **route-reflector-client**
14. **exit-address-family**
15. **address-family ipv4 vrf** *vrf-name*
16. **no synchronization**
17. **exit-address-family**
18. **address-family ipv4 multicast vrf** *vrf-name*
19. **no synchronization**
20. **exit-address-family**
21. **end**
22. **show running-config | b router bgp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf1* **Example:**	Defines a VRF instance and enters VRF configuration mode.

	Command or Action	Purpose
	`Device(config)# vrf definition vrf1`	
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 1:1`	Creates a route target export extended community for a VRF instance.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 1:1`	Creates a route target import extended community for a VRF instance.
Step 7	**address-family ipv4** **Example:** `Device(config-router)# address-family ipv4`	Configures a routing session using IPv4 address prefixes and enters address family configuration mode.
Step 8	**mdt default** *group-address* **Example:** `Device(config-vrf)# mdt default 239.0.0.1`	Configures a default multicast distribution tree (MDT) group for a VRF instance.
Step 9	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 10	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Configures the BGP routing process and enters router configuration mode.
Step 11	**address-family vpnv4 multicast** **Example:** `Device(config-router)# address-family vpnv4 multicast`	Configures a routing session using VPN Version 4 multicast address prefixes and enters address family configuration mode.
Step 12	**neighbor** *peer-group-name* **send-community extended** **Example:**	Specifies that a communities attribute should be sent to a BGP neighbor.

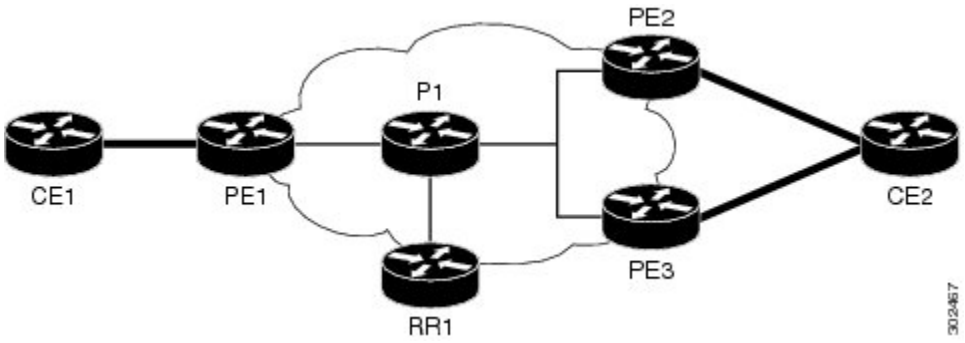
	Command or Action	Purpose
	`Device(config-router-af)# neighbor client1 send-community extended`	
Step 13	**neighbor** *peer-group-name* **route-reflector-client** **Example:** `Device(config-router-af)# neighbor client1 route-reflector-client`	(Optional) Configures the router as a BGP route reflector and configures the specified neighbor as its client.
Step 14	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 15	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf vrf1`	Places the router in address family configuration mode and specifies the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 16	**no synchronization** **Example:** `Device(config-router-af)# no synchronization`	Enables the Cisco software to advertise a network route without waiting for the Interior Gateway Protocol (IGP) system.
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 18	**address-family ipv4 multicast vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 multicast vrf vrf1`	Configures a routing session using IPv4 multicast address prefixes for a VRF instance and enters address family configuration mode.
Step 19	**no synchronization** **Example:** `Device(config-router-af)# no synchronization`	Enables the Cisco software to advertise a network route without waiting for the IGP system.
Step 20	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters router configuration mode.
Step 21	**end** **Example:**	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 22	**show running-config | b router bgp** **Example:** `Device# show running-config | b router bgp`	(Optional) Displays the running configuration for specified device.

Configuration Examples for BGP--mVPN BGPsAFI 129 - IPv4

Example: Configuring BGP - mVPN BGP sAFI 129 - IPv4

This example uses the topology illustrated in the figure below.



The following example configures BGP SAFI 129 on the route reflector (RR):

```
!
ip multicast-routing
!
!<<< Define BGP update-source loopback0
!<<< on RR as 192.0.2.10
interface loopback0
 ip pim sparse-dense-mode
 ip address 192.0.2.10 255.255.255.255
!
.
.
.
router bgp 65000
 no synchronization
 neighbor 192.0.2.1 remote-as 65000
 neighbor 192.0.2.1 update-source loopback0
 neighbor 192.0.2.2 remote-as 65000
 neighbor 192.0.2.2 update-source loopback0
 neighbor 192.0.2.3 remote-as 65000
 neighbor 192.0.2.3 update-source loopback0
 !
 .
 .
 address-family vpnv4 unicast
  neighbor 192.0.2.1 activate
```

```
  neighbor 192.0.2.1 send-community extended
  neighbor 192.0.2.1 route-reflector-client
  neighbor 192.0.2.2 activate
  neighbor 192.0.2.2 send-community extended
  neighbor 192.0.2.2 route-reflector-client
  neighbor 192.0.2.3 activate
  neighbor 192.0.2.3 send-community extended
  neighbor 192.0.2.3 route-reflector-client
  exit-address-family
 !
 address-family vpnv4 multicast
  !<<< want route from CE1 with nexthop
  !<<< through PE3 in multicast routing table
  neighbor 192.0.2.1 activate
  neighbor 192.0.2.1 send-community extended
  neighbor 192.0.2.1 route-reflector-client
  neighbor 192.0.2.3 activate
  neighbor 192.0.2.3 send-community extended
  neighbor 192.0.2.3 route-reflector-client
 exit-address-family
!
.
.
```

The following example configures BGP SAFI 129 on the PE1 router (PE2 and PE3 will have a similar configuration):

```
Hostname PE1
!
vrf definition vrf1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 !
 address-family ipv4
  mdt default 239.0.0.1
 exit-address-family
!
ip multicast-routing
ip multicast-routing vrf vrf1
!
.
.
.
!<<< Define BGP update-source on Loopback0
!<<< on PE1
inteface loopback0
 ip pim sparse-dense-mode
 ip address 192.0.2.1 255.255.255.255
!
.
.
.
!<<< Define vrf vrf1 interface on PE1 to CE1
interface ethernet0/0
 vrf forwarding vrf1
 ip pim sparse-dense-mode
 ip address 192.0.2.1 255.255.255.0
!
.
.
,
router bgp 65000
```

```
 !<<<< PE peer neighbor with RR
 neighbor 192.0.2.10 remote-as 65000
 neighbor 192.0.2.10 update-source loopback0
 no synchronization
 .
 .
 .
 address-family vpnv4
  neighbor 192.0.2.10 activate
  neighbor 192.0.2.10 send-community extended
 exit-address-family
 !
 !<<< Define vpnv4 safi129 with neighbor
 !<<< to RR
 address-family vpnv4 multicast
  neighbor 192.0.2.10 activate
  neighbor 192.0.2.10 send-community extended
 exit-address-family
 !
 .
 .
 .
 !<<< Define unicast address-family vrf vrf1.
 !<<< PE-CE is eBGP in this case.
 !<<< If PE-CE is not eBGP, please use
 !<<< redistribute cli, instead of
 !<<< neighbor cli below.
 address-family ipv4 vrf vrf1
  no synchronization
  redistribute connected
  neighbor 192.0.2.5 remote-as 65011
 exit-address-family
 !
 !<<< Define multicast address-family vrf vrf1
 !<<< (safi2. PE-CE is eBGP in this case.
 !<<< If PE-CE is not eBGP, please use
 !<<< redistribute cli, instead of
 !<<< neighbor cli below.
 address-family ipv4 multicast vrf vrf1
  no synchronization
  redistribute connected
  neighbor 192.0.2.5 remote-as 65011
 exit-address-family
 !
```

The following example configures BGP SAFI 129 on the CE1 router. (In this case, PE-CE routing is eBGP. CE2 will have a similar configuration):

```
interface ethernet0/0
 ip address 192.0.2.5 255.255.255.0
 ip pim sparse-dense-mode
!
.
.
.
router bgp 65011
 bgp router-id 192.0.2.5
 bgp log-neighbor-changes
 !
 address-family ipv4
  redistribute connected
  neighbor 192.0.2.1 remote-as 65000
 exit-address-family
```

```
 !
 address-family ipv4 multicast
  redistribute connected
  neighbor 192.0.2.1 remote-as 65000
 exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP - mVPN BGP sAFI 129 - IPv4

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 128: Feature Information for BGP - mVPN BGP sAFI 129 - IPv4

Feature Name	Releases	Feature Information
BGP - mVPN BGP sAFI 129 - IPv4	15.2(2)S 15.2(4)S Cisco IOS XE Release 3.6S	The BGP - mVPN BGP sAFI 129 IPv4 feature provides the capability to support multicast routing in the service provider's core IPv4 network. This feature is needed to support BGP-based MVPNs. BGP MVPN provides a means for service providers to use different encapsulation methods (generic route encapsulation (GRE), Multicast Label Distribution Protocol (MLDP), and ingress replication) for forwarding MVPN multicast data traffic in the service provider network. In Cisco IOS Release 15.2(4)S, support was added for the Cisco 7200 series router. The following commands were modified: **address-family ipv4**, **address-family vpnv4**.

CHAPTER 100

BGP-MVPN SAFI 129 IPv6

Subsequent Address Family Identifier (SAFI) 129, known as VPN Multicast SAFI, provides the capability to support multicast routing in the service provider's core IPv6 network.

Border Gateway Protocol (BGP) Multicast Virtual Private Network (MVPN) provides a means for service providers to use different encapsulation methods (generic routing encapsulation [GRE], Multicast Label Distribution Protocol [MLDP], and ingress replication) for forwarding MVPN multicast data traffic in the service provider network.

The BGP-MVPN SAFI 129 IPv6 feature is required to support BGP-based MVPNs.

Prerequisites for BGP-MVPN SAFI 129 IPv6

- Before you configure a SAFI 129 IPv6-related address family, the **ipv6 unicast-routing** command must be configured on the device.
- To create a multicast IPv6 VRF address family under BGP, IPv6 must first be activated on the VRF itself.

Note There is no separate multicast configuration on the VRF. Configuring the **address-family ipv6** command on the VRF will enable both unicast and multicast topologies.

- If you want prefixes to be installed into the Routing Information Base (RIB), you must configure the **pim** command on a VRF interface.

Information About BGP-MVPN SAFI 129 IPv6

Overview of BGP-MVPN SAFI 129 IPv6

MVPN utilizes the existing VPN infrastructure to allow multicast traffic to pass through the provider space. Information derived from VPN routes is one of the components needed to set up tunnels within the core. Currently, multicast traffic will derive this information from the unicast VPNv6 tables, which forces multicast traffic to be dependent on unicast topologies.

For scenarios in which multicast and unicast traffic would be better suited with separate topologies, the customer edge (CE) router may advertise a special set of routes to be used exclusively for multicast VPNs. Multicast routes learned from the CE router can be propagated to remote provider edge (PE) routers via SAFI 129. Multicast routes learned from the CE router or multicast VPN routes learned from remote PE routers can now be installed directly into the multicast RIB, instead of using replicated routes from the unicast RIB. Maintaining separate routes and entries for unicast and multicast allows you to create differing topologies for each service within the core.

How to Configure BGP-MVPN SAFI 129 IPv6

Configuring BGP-MVPN SAFI 129 IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf1*
4. **rd** *route-distinguisher*
5. **route-target export** *route-target-ext-community*
6. **route-target import** *route-target-ext-community*
7. **address-family ipv6**
8. **mdt default** *group-address*
9. **exit**
10. **exit**
11. **router bgp** *autonomous-system-number*
12. **address-family vpnv6 multicast**
13. **neighbor** *peer-group-name* **send-community extended**
14. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate**
15. **address-family ipv6 multicast vrf** *vrf-name*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf1* **Example:** `Device(config)# vrf definition vrf1`	Defines a VRF instance and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	Specifies a route distinguisher (RD) for a VRF instance.
Step 5	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 1:1`	Creates a route target export extended community for a VRF instance.
Step 6	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 1:1`	Creates a route target import extended community for a VRF instance.
Step 7	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Configures a routing session using IPv6 address prefixes and enters address family configuration mode.
Step 8	**mdt default** *group-address* **Example:** `Device(config-vrf-af)# mdt default 239.0.0.1`	Configures a default multicast distribution tree (MDT) group for a VRF instance.
Step 9	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits address family configuration mode and enters VRF configuration mode.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 11	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Configures a BGP routing process and enters router configuration mode.
Step 12	**address-family vpnv6 multicast** **Example:** `Device(config-router)# address-family vpnv6 multicast`	Configures a routing session using VPN Version 6 multicast address prefixes and enters address family configuration mode.
Step 13	**neighbor** *peer-group-name* **send-community extended** **Example:** `Device(config-router-af)# neighbor client1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 14	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address* **%**} **activate** **Example:** `Router(config-router-af)# neighbor 2001:DB8:0:CC00::1 % activate`	Enables the neighbor to exchange prefixes for the specified family type with the neighbor and the local router.
Step 15	**address-family ipv6 multicast vrf** *vrf-name* **Example:** `Device(config-router-af)# address-family ipv6 multicast vrf vrf1`	Configures a routing session using IPv6 multicast address prefixes for a VRF instance.
Step 16	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP-MVPN SAFI 129 IPv6

Example: Configuring BGP-MVPN SAFI 129 IPv6

The example below shows the configuration for a PE router:

```
hostname PE1
!
!
vrf definition blue
 rd 55:1111
 route-target export 55:1111
 route-target import 55:1111
 !
 address-family ipv6
  mdt default 232.1.1.1
  mdt data 232.1.200.0 0.0.0.0
 exit-address-family
!
!ip multicast-routing
ip multicast-routing vrf blue
ip cef
!
ipv6 unicast-routing
ipv6 multicast-routing
ipv6 multicast-routing vrf blue
ipv6 cef
!
!interface Loopback0
 ip address 205.1.0.1 255.255.255.255
 ip pim sparse-dense-mode
 ipv6 address FE80::205:1:1 link-local
 ipv6 address 205::1:1:1/64
 ipv6 enable
!
interface Ethernet0/0
 ! interface connect to the core vpn
 bandwidth 1000
 ip address 30.3.0.1 255.255.255.0
 ip pim sparse-dense-mode
 delay 100
 ipv6 address FE80::70:1:1 link-local
 ipv6 address 70::1:1:1/64
 ipv6 enable
 mpls ip
!
interface Ethernet1/1
 ! interface connect to CE (vrf interface)
 bandwidth 1000
 vrf forwarding blue
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-dense-mode
 delay 100
 ipv6 address FE80::20:1:1 link-local
 ipv6 address 20::1:1:1/64
 ipv6 enable
!
router ospf 200
 redistribute connected subnets
 redistribute bgp 55 metric 10
 passive-interface Loopback0
 network 30.3.0.0 0.0.255.255 area 1
!
router bgp 55
 bgp log-neighbor-changes
 no bgp default route-target filter
 ! neighbor to another PE in core
 neighbor 205.3.0.3 remote-as 55
 neighbor 205.3.0.3 update-source Loopback0
```

```
 !
 address-family ipv4 mdt
  ! neighbor to another PE in core
  neighbor 205.3.0.3 activate
  neighbor 205.3.0.3 send-community extended
 exit-address-family
 !
 address-family vpnv6
  ! neighbor to another PE in core
  neighbor 205.3.0.3 activate
  neighbor 205.3.0.3 send-community extended
 exit-address-family
 !
 address-family vpnv6 multicast
  ! neighbor to another PE in core
  ! this address-family is added to enable
  ! safi129 between two PEs
  neighbor 205.3.0.3 activate
  neighbor 205.3.0.3 send-community extended
 exit-address-family
 !
 address-family ipv6 vrf blue
  ! neighbor to CE1 in vrf
  redistribute connected
  redistribute static
  neighbor FE80::20:1:6%Ethernet1/1 remote-as 56
  neighbor FE80::20:1:6%Ethernet1/1 activate
 exit-address-family
 !
 address-family ipv6 multicast vrf blue
  !  neighbor to CE1 in vrf
  ! this address-family is added to enable
  ! safi2 on PE-CE
  redistribute connected
  redistribute static
  neighbor FE80::20:1:6%Ethernet1/1 remote-as 56
  neighbor FE80::20:1:6%Ethernet1/1 activate
 exit-address-family
!
ipv6 pim vrf blue rp-address 201::1:1:7 blue_bidir_acl bidir
ipv6 pim vrf blue rp-address 202::1:1:6 blue_sparse_acl
!
ipv6 access-list black_bidir_acl
 permit ipv6 any FF06::/64
!
ipv6 access-list black_sparse_acl
 permit ipv6 any FF04::/64
!
ipv6 access-list blue_bidir_acl
 permit ipv6 any FF05::/64
!
ipv6 access-list blue_sparse_acl
 permit ipv6 any FF03::/64
!
end
```

The example below shows the configuration for a CE router:

```
hostname CE1
!
ip multicast-routing
ip cef
ipv6 unicast-routing
```

```
ipv6 multicast-routing
ipv6 multicast rpf use-bgp
ipv6 cef
!
interface Ethernet1/1
 bandwidth 1000
 ip address 10.1.0.6 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim sparse-dense-mode
 delay 100
 ipv6 address FE80::20:1:6 link-local
 ipv6 address 20::1:1:6/64
 ipv6 enable
 no keepalive
!
router bgp 56
 bgp log-neighbor-changes
 neighbor FE80::20:1:1%Ethernet1/1 remote-as 55
 !
 address-family ipv6
  redistribute connected
  redistribute static
  neighbor FE80::20:1:1%Ethernet1/1 activate
 exit-address-family
 !
 address-family ipv6 multicast
  redistribute connected
  redistribute static
  neighbor FE80::20:1:1%Ethernet1/1 activate
 exit-address-family
!
ipv6 pim rp-address 201::1:1:7 blue_bidir_acl bidir
ipv6 pim rp-address 202::1:1:6 blue_sparse_acl
!
ipv6 access-list blue_bidir_acl
 permit ipv6 any FF05::/64
!
ipv6 access-list blue_sparse_acl
 permit ipv6 any FF03::/64
!
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
BGP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference*

Standards and RFCs

Standard/RFC	Title
MDT SAFI	*Subsequent Address Family Identifiers (SAFI) Parameters*
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP-MVPN SAFI 129 IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 129: Feature Information for BGP—MVPN SAFI 129 IPv6

Feature Name	Releases	Feature Information
BGP—MVPN SAFI 129 IPv6	15.2(4)S Cisco IOS XE Release 3.7S 15.3(1)T	SAFI 129 , known as VPN Multicast SAFI, provides the capability to support multicast routing in the service provider's core IPv6 network. The following commands were introduced or modified: **address-family ipv6**, **address-family vpnv6**, and **show bgp vpnv6 multicast**.

CHAPTER 101

BFD—BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

The BFD—BGP Multihop Client Support feature enables Border Gateway Protocol (BGP) to use multihop Bidirectional Forwarding Detection (BFD) support, which improves BGP convergence as BFD detection and failure times are faster than the Interior Gateway Protocol (IGP) convergence times in most network topologies.

The BFD—BGP cBIT feature allows BGP to determine if BFD failure is dependent or independent of the Control Plane. This allows BGP greater flexibility in handling BFD down events.

Restrictions for BFD—BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

- For BGP IPv4 and BGP IPv6 peering sessions only, multihop BFD support is available for BGP for address-family IPv4 and IPv6 unicast.
- For multihop BGP sessions using IPv6 Link Local addresses, BFD multihop support is not available.
- Currently BFD Hardware offload is not supported for multihop BFD sessions and so C-bit will not be set for multihop sessions.
- Multihop BFD for IPv6 Virtual Routing and Forwarding (VRF) is not supported.
- BGP session attribute for BFD does not change dynamically when BGP session changes from single-hop to multihop, hence you need to clear the existing BGP session to reinitiate multihop BFD session.

Information About BFD - BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

BFD—BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

BFD is a detection protocol that is designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning is easier, and reconvergence time is consistent and predictable. The main benefit of implementing BFD for BGP is a significantly faster reconvergence time. For internal BGP (iBGP) sessions and external BGP (eBGP) sessions that are either single hop or multihop, BGP can use of the multihop BFD support to help improve the BGP convergence because BFD detection and failure times are faster than the IGP convergence times in most of the network topologies. BGP needs the support of multihop BFD as described in RFC5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*.

BGP by default will purge the routes received from a specific peer when a BFD down event occurs and BFD informs BGP about it. The cBit in BFD determines whether BFD is dependent or independent of the Control Plane. Clients like BGP, whose peers are enabled with fast fall over feature with BFD support, can use this BFD cBit support to provide a more deterministic mechanism to do nonstop forwarding (NSF) when BGP graceful restart is enabled along with BFD fast-fallover support for BGP sessions.

When BGP is using BFD for the fast fallover feature for remote connectivity detection, BFD can detect some of those failures. If BFD is independent of the control plane, a BFD session failure means that data cannot be forwarded anymore (due to link control failures) and so the BGP graceful restart procedures should be terminated to avoid null routes. On the other hand, when BFD is dependent on the control plane, a BFD failure cannot be separated out from the other events taking place in the control plane. When the control plane crashes, a switchover happens and BFD restarts. It is best for the clients (like BGP) to avoid any terminations due to the graceful restart taking place.

The table below describes the handling of BFD down events by BGP.

Table 130: BGP handling of BFD Down Event

BFD Down Event	Failure—Control Plane Independent?	BGP Action for NSF (when GR and BFD are enabled)
BGP control plane failure detection enabled	Yes	Purge Routes
BGP control plane failure detection enabled	No	Carry on NSF and keep stale routes in Routing Information Base (RIB)
BGP control plane failure detection disabled (the default behavior)	Yes	Purge Routes
BGP control plane failure detection disabled (the default behavior)	No	Purge Routes

BGP session establishment works independently from BFD state change, except for fast fall-over detection, that is, inaccessible next-hop and cause best path re-calculation. This means that the BGP session could be established while BFD state is down or dampened, even with neighbor fail-over bfd configured.

From the XE 3.17S release the new optional keyword strict-mode is introduced, which does not allow BGP session to become established, if BFD is in down state. When BFD is dampened or down the routing protocol states or sessions cannot come up.

How to Configure BFD - BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

Configuring BFD—BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

Before you begin

Note The multihop BFD minimum detection time should be higher than IGP convergence times in your network to ensure that down events are not mistakenly identified during reconvergences, causing multihop BGP sessions to flap.

Note For the BFD strict mode to work, configure BFD on both the neighboring devices.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router bgp** *autonomous-system-number*

Example:

```
Device(config)# router bgp 50000
```

Configures the Border Gateway Protocol (BGP) routing process and enters router configuration mode.

Step 4 **neighbor** *ip-address* **remote-as** *autonomous-system-number*

Example:

```
Device(config-router)# neighbor 10.0.0.2 remote-as 100
```

Adds an entry to the BGP or multiprotocol BGP neighbor table.

Step 5 **neighbor** *ip-address* **update-source** *interface-type interface-number*

Example:

```
Device(config-router)# neighbor 10.0.0.2 update-source GigabitEthernet 0/0/0
```

Allows BGP sessions to use any operational interface for TCP connections.

Step 6 **neighbor** *ip-address* **remote-as** *autonomous-system-number*

Example:

```
Device(config-router)# neighbor 10.0.0.2 remote-as 100
```

Adds an entry to the BGP or multiprotocol BGP neighbor table.

Step 7 **neighbor** *ip-address* **ebgp-multihop** *ttl*

Example:

```
Device(config-router)# neighbor 10.0.0.2 ebgp-multihop 4
```

Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.

Step 8 **neighbor** *ip-address* **fall-over bfd** [**multi-hop**] [**check-control-plane-failure**] [**strict-mode**]

Example:

```
Device(config-router)# neighbor 10.0.0.2 fall-over bfd multi-hop check-control-plane-failure strict-mode
```

- Enables BGP to monitor the peering session of a specified neighbor for adjacency changes and to deactivate the peering session.
- Configures BGP BFD with control plane independence that is enabled for BFD cBit support.

Note When **check-control-plane-failure** is enabled for the neighbor and the trigger for adjacency down is set at BFD-Down notification, then BGP looks into C-Bit to decide whether to carry out Cisco Nonstop Forwarding or Purge the routes. BGP looks into C-Bit in BFD packet and decides only when **check-control-plane-failure** is enabled.

Step 9 **end**

Example:

```
Device(config-router)# end
```

Exits router configuration mode and returns to privileged EXEC mode.

Configuration Examples for BFD - BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

Example: Configuring BFD—BGP Multihop Client Support, cBit (IPv4/IPv6), and Strict Mode

```
R1 e0/0 ---------------e0/0 R2

Router 1 configuration

hostname R1
!
bfd map ipv4 2.2.2.2/32 1.1.1.1/32 mh1
!
bfd-template multi-hop mh1
interval min-tx 200 min-rx 200 multiplier 3
!
interface Loopback1
ip address 1.1.1.1 255.255.255.255
ip ospf 1 area 0
!
interface Ethernet0/0
ip address 10.0.0.1 255.255.255.0
ip ospf 1 area 0
!
router ospf 1
!
router bgp 1
neighbor 2.2.2.2 remote-as 1
neighbor 2.2.2.2 update-source Loopback1
neighbor 2.2.2.2 fall-over bfd multi-hop check-control-plane-failure strict-mode
!
address-family ipv4
neighbor 2.2.2.2 activate
exit-address-family
!

Router 2 configuration:

hostname R2
!
bfd map ipv4 1.1.1.1/32 2.2.2.2/32 mh1
bfd-template multi-hop mh1
interval min-tx 200 min-rx 200 multiplier 3
!
interface Loopback1
ip address 2.2.2.2 255.255.255.255
ip ospf 1 area 0
!
interface Ethernet0/0
```

```
ip address 10.0.0.2 255.255.255.0
ip ospf 1 area 0
!
router ospf 1
!
router bgp 1
neighbor 1.1.1.1 remote-as 1
neighbor 1.1.1.1 update-source Loopback1
neighbor 1.1.1.1 fall-over bfd multi-hop check-control-plane-failure strict-mode
!
address-family ipv4
neighbor 1.1.1.1 activate
exit-address-family
!
```

Verifying BFD—BGP Multihop Client Support, cBit (IPv4 and IPv6), and Strict Mode

The following examples show how to verify if BFD is enabled on the neighbor, peer-group.

```
R801-ASBR#sh ip bgp neighbor 10.1.0.2
BGP neighbor is 10.1.0.2,  remote AS 65000, external link
 Fall over configured for session
BFD is configured. BFD peer is Up. Using BFD to detect fast fallover (single-hop)
 in strict-mode.
  BGP version 4, remote router ID 10.10.10.10
  BGP state = Established, up for 00:04:12
  Last read 00:00:49, last write 00:00:24, hold time is 180, keepalive interval
is 60 seconds
…
```

If BFD is up and running, the following is displayed:

```
Fall over configured for session
BFD is configured. BFD peer is Up. Using BFD to detect fast fallover (single-hop)
 in strict-mode (will be verified).
…
```

If BFD is not up and running, the following is displayed:

```
Fall over configured for session
 BFD is configured. BFD peer is Down. Using BFD to detect fast fallover
(single-hop) in strict-mode.
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BFD—BGP Multihop Client Support, cBit (IPv4/IPv6), and Strict Mode

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 131: Feature Information for BFD—BGP Multihop Client Support, cBit (IPv4/IPv6), and Strict Mode

Feature Name	Releases	Feature Information
BFD—BGP Multihop Client Support and cBit (IPv4/IPv6)	15.2(4)S Cisco IOS XE Release 3.6S Cisco IOS XE Release 3.7S	The BFD—BGP Multihop Client Support feature enables Border Gateway Protocol (BGP) to use multihop Bidirectional Forwarding Detection (BFD) support, which improves BGP convergence as BFD detection and failure times are faster than the Interior Gateway Protocol (IGP) convergence times in most of network topologies. The BFD—BGP cBIT feature allows BGP to determine if BFD failure is dependent or independent of the Control Plane. This allows BGP greater flexibility in handling BFD down events. In Cisco IOS XE Release 3.7S, support was added for the Cisco ASR 903 router. The following commands were modified: **neighbor fall-over** and **show ip bgp neighbors**.
BFD—BGP Multihop Client Support, cBit (IPv4/IPv6), and Strict Mode	Cisco IOS XE Release 3.17S	In Cisco IOS XE Release 3.17S, the following command was modified: **neighbor** *ip-address* **fall-over bfd** [**multi-hop**|**single-hop**] [**check-control-plane-failure**] [**strict-mode**] .

CHAPTER 102

BGP Attribute Filter and Enhanced Attribute Error Handling

The BGP Attribute Filter feature allows you to "treat-as-withdraw" updates that contain specific path attributes. The prefixes contained in the update are removed from the routing table. The feature also allows you to remove specific path attributes from incoming updates. Both behaviors provide an increased measure of security. The BGP Enhanced Attribute Error Handling feature prevents peer sessions from flapping due to errors from any malformed update, thereby saving resources.

Information About BGP Attribute Filtering

BGP Attribute Filter and Enhanced Attribute Error Handling

The BGP Attribute Filter feature provides two ways to achieve an increased measure of security:

- The feature allows you to treat-as-withdraw an Update coming from a specified neighbor if the Update contains a specified attribute type. When an Update is treat-as-withdraw, the prefixes in the Update are removed from the BGP routing table (if they existed in the routing table).
- The feature also allows you to drop specified path attributes from an Update, and then the system processes the rest of the Update as usual.

The BGP Enhanced Attribute Error Handling feature prevents peer sessions from flapping due to a malformed Update. The malformed Update is treat-as-withdraw and does not cause the BGP session to be reset. This feature is enabled by default, but can be disabled.

The features are implemented in the following order:

1. Received Updates that contain user-specified path attributes are treat-as-withdraw (as long as the NLRI can be parsed successfully). If there is an existing prefix in the BGP routing table, it will be removed. The **neighbor path-attribute treat-as-withdraw** command configures this feature.

2. User-specified path attributes are discarded from received Updates, and the rest of the Update is processed normally. The **neighbor path-attribute discard** command configures this feature.
3. Received Updates that are malformed are treat-as-withdraw. This feature is enabled by default; it can be disabled by configuring the **no bgp enhanced-error** command.

Details About Specifying Attributes as Treat-as-Withdraw

Attribute types 1, 2, 3, 4, 8, 14, 15, and 16 cannot be configured for path attribute treat-as-withdraw.

Attribute type 5 (localpref), type 9 (Originator,) and type 10 (Cluster-id) can be configured for treat-as-withdraw for eBGP neighbors only.

Configuring path attributes to be treated as withdrawn will trigger an inbound Route Refresh to ensure that the routing table is up to date.

Details About Specifying Attributes as Discard

Attribute types 1, 2, 3, 4, 8, 14, 15, and 16 cannot be configured for path attribute discard.

Attribute type 5 (localpref), type 9 (Originator), and type 10 (Cluster-id) can be configured for discard for eBGP neighbors only.

Configuring path attributes to be discarded will trigger an inbound Route Refresh to ensure that the routing table is up to date.

Details About Enhanced Attribute Error Handling

If a malformed Update is received, it is treat-as-withdraw to prevent peer sessions from flapping due to the processing of BGP path attributes. This feature applies to eBGP and iBGP peers. This feature is enabled by default; it can be disabled.

If the BGP Enhanced Attribute Error Handling feature is enabled or disabled, BGP places the MP_REACH attribute (attribute 14) at the beginning of an attribute list while formatting an update. Enhanced attribute error handling functions more easily when the MP_REACH attribute is at the beginning of the attribute list.

How to Filter BGP Path Attributes

Treat-as-Withdraw BGP Updates Containing a Specified Path Attribute

Note Performing this task will trigger an inbound Route Refresh to ensure that the routing table is up to date.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address*} **path-attribute treat-as-withdraw** {*attribute-value* | **range** *start-value end-value*} **in**
5. Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode.
Step 4	**neighbor** {*ip-address* | *ipv6-address*} **path-attribute treat-as-withdraw** {*attribute-value* | **range** *start-value end-value*} **in** **Example:** `Device(config-router)# neighbor 2001:DB8:1::1 path-attribute treat-as-withdraw 100 in`	Treat-as-withdraw any incoming Update messages that contain the specified path attribute or range of path attributes. • Any prefixes in an Update that is treat-as-withdraw are removed from the BGP routing table. • The specific attribute value and the range of attribute values are independent of each other.
Step 5	Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor.	
Step 6	**end** **Example:** `Device(config-router)# end`	Exits to privileged EXEC mode.

Discarding Specific Path Attributes from an Update Message

Note Performing this task will trigger an inbound Route Refresh to ensure that the routing table is up to date.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address*} **path-attribute discard** {*attribute-value* | **range** *start-value end-value*} **in**

5. Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 6500`	Configures a BGP routing process and enters router configuration mode.
Step 4	**neighbor** {*ip-address* | *ipv6-address*} **path-attribute discard** {*attribute-value* | **range** *start-value end-value*} **in** **Example:** `Device(config-router)# neighbor 2001:DB8:1::1 path-attribute discard 128 in`	Drops specified path attributes from Update messages from the specified neighbor.
Step 5	Repeat Step 4 to configure other attributes not in a range or to configure a different neighbor. **Example:**	
Step 6	**end** **Example:** `Device(config-router)# end`	Exits to privileged EXEC mode.

Displaying Withdrawn or Discarded Path Attributes

Perform any of these steps in any order to display information about treat-as-withdraw, discarded, or unknown path attributes. You can use the **show ip bgp** command with any address family that BGP supports, such as **show ip bgp ipv4 multicast**, **show ip bgp ipv6 unicast**, etc.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbor** [*ip-address* | *ipv6-address*]
3. **show ip bgp path-attribute unknown**

4. **show ip bgp path-attribute discard**
5. **show ip bgp vpnv4 all** *prefix*
6. **show ip bgp neighbors** *prefix*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp neighbor** [*ip-address* | *ipv6-address*] **Example:** `Device# show ip bgp neighbor 2001:DB8:1::1`	(Optional) Displays the configured discard and treat-as-withdraw attribute values for the neighbor, counts of Updates with such attributes discarded or treat-as-withdraw, and the count of malformed treat-as-withdraw Updates.
Step 3	**show ip bgp path-attribute unknown** **Example:** `Device# show ip bgp path-attribute unknown`	(Optional) Displays all prefixes that have an unknown attribute.
Step 4	**show ip bgp path-attribute discard** **Example:** `Device# show ip bgp path-attribute discard`	(Optional) Displays all prefixes for which an attribute has been discarded.
Step 5	**show ip bgp vpnv4 all** *prefix* **Example:** `Device# show ip bgp vpnv4 all 192.168.1.0`	(Optional) Displays the unknown attributes and discarded attributes associated with a prefix.
Step 6	**show ip bgp neighbors** *prefix* **Example:** `Device# show ip bgp neighbors 192.168.1.0`	(Optional) Displays the configured discard and treat-as-withdraw attributes associated with a prefix.

Configuration Examples for BGP Attribute Filter

Examples: Withdraw Updates Based on Path Attribute

The following example shows how to configure the device to treat-as-withdraw any Update messages from the specified neighbor that contain the unwanted path attribute 100 or 128:

```
router bgp 65600
 neighbor 2001:DB8:1::2 path-attribute treat-as-withdraw 100 in
```

```
neighbor 2001:DB8:1::2 path-attribute treat-as-withdraw 128 in
```

The following example shows how to configure the device to treat-as-withdraw any Update messages from the specified neighbor that contain the unwanted path attributes in the range from 21 to 255:

```
router bgp 65600
 neighbor 2001:DB8:1::2 path-attribute treat-as-withdraw 21 255 in
```

Examples: Discard Path Attributes from Updates

The following example shows how to configure the device to discard path attributes 100 and 128 from incoming Update messages from the specified neighbor. The rest of the Update message will be processed as usual.

```
router bgp 65600
 neighbor 2001:DB8:1::1 path-attribute discard 100 in
 neighbor 2001:DB8:1::1 path-attribute discard 128 in
```

The following example shows how to configure the device to discard path attributes in the range from 17 to 255 from incoming Update messages from the specified neighbor. The rest of the Update message will be processed as usual.

```
router bgp 65600
 neighbor 2001:DB8:1::1 path-attribute discard 17 255 in
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
draft-ietf-idr-error-handling	Revised Error Handling for BGP Updates from External Neighbors

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Attribute Filter and Enhanced Attribute Error Handling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 132: Feature Information for BGP Attribute Filter and Enhanced Attribute Error Handling

Feature Name	Releases	Feature Information
BGP Attribute Filter and Enhanced Attribute Error Handling		The BGP Attribute Filter allows you to "treat-as-withdraw" updates that contain specific path attributes. The prefixes contained in the update are removed from the routing table. The feature also allows you to remove specific path attributes from incoming updates. Both behaviors provide an increased measure of security. The BGP Enhanced Attribute Error Handling feature prevents peer sessions from flapping due to errors from any malformed update, thereby saving resources. The following commands were introduced: **bgp enhanced-error**, **neighbor path-attribute discard**, **neighbor path-attribute treat-as-withdraw**, **show ip bgp path-attribute discard**, and **show ip bgp path-attribute unknown**. The following commands were modified: **show ip bgp**, **show ip bgp neighbor**, and **show ip bgp vpnv4 all**.

CHAPTER 103

BGP Additional Paths

The BGP Additional Paths feature allows the advertisement of multiple paths through the same peering session for the same prefix without the new paths implicitly replacing any previous paths. This behavior promotes path diversity and reduces multi-exit discriminator (MED) oscillations.

Information About BGP Additional Paths

Problem That Additional Paths Can Solve

BGP routers and route reflectors (RRs) propagate only their best path over their sessions. The advertisement of a prefix replaces the previous announcement of that prefix (this behavior is known as an implicit withdraw). The implicit withdraw can achieve better scaling, but at the cost of path diversity.

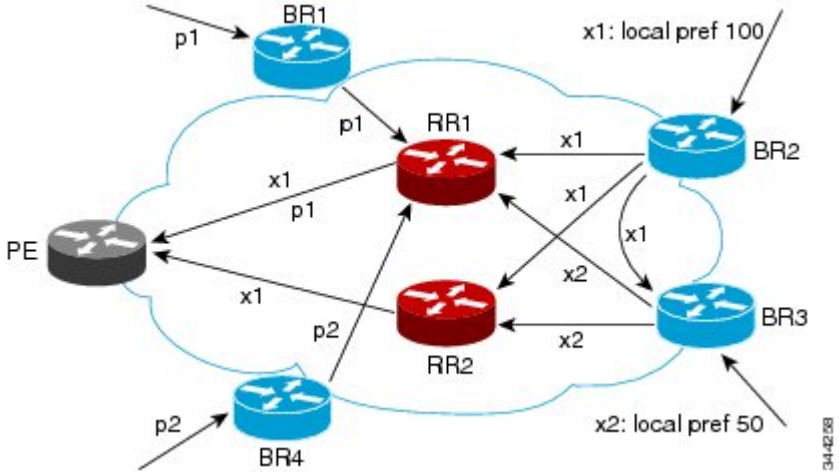
Path hiding can prevent efficient use of BGP multipath, prevent hitless planned maintenance, and can lead to MED oscillations and suboptimal hot-potato routing. Upon nexthop failures, path hiding also inhibits fast and local recovery because the network has to wait for BGP control plane convergence to restore traffic. The BGP Additional Paths feature provides a generic way of offering path diversity; the Best External or Best Internal features offer path diversity only in limited scenarios.

The BGP Additional Paths feature provides a way for multiple paths for the same prefix to be advertised without the new paths implicitly replacing the previous paths. Thus, path diversity is achieved instead of path hiding.

Path-Hiding Scenario

This section describes in more detail how path hiding can occur. In the following figure, path p1 is advertised from BR1, and path p2 is advertised from BR4, to RR1. RR1 selects the best path of the two and advertises path p1 to PE.

Figure 117: RR Hiding an Additional Path

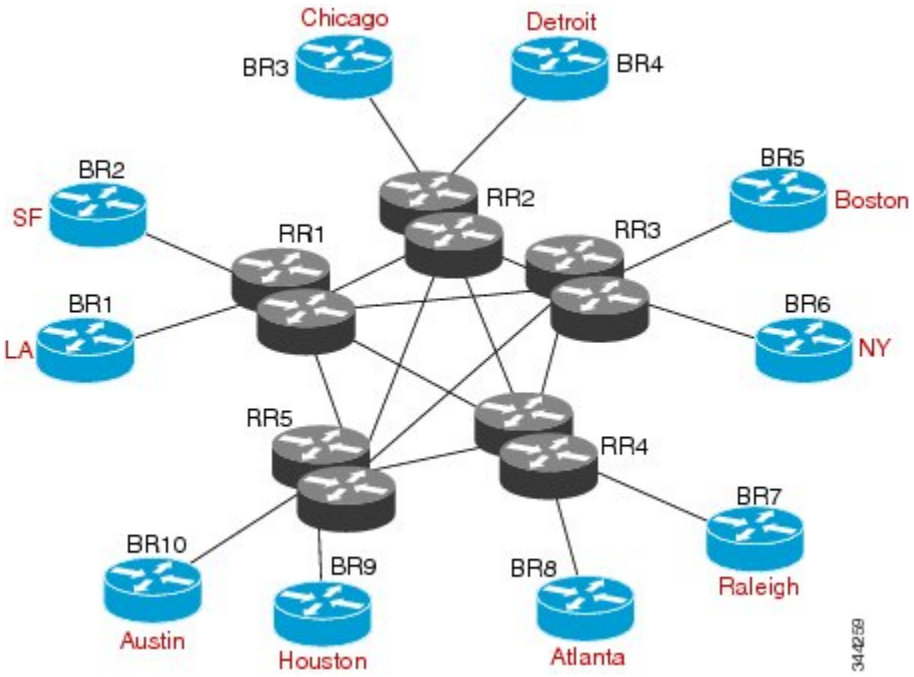


In the figure above, we also see prefix x with path x1 being advertised from BR2 to BR3 (which has path x2) with local preference 100. BR3 also has path x2, but due to routing policy, BR3 will advertise to the RRs x1 (not shown) instead of x2, and x2 will be suppressed. A user could enable the advertisement of best external route on BR3 and thereby advertise x2 to the RRs, but, again, the RRs advertise only the best path.

Suboptimal Hot-Potato Routing Scenario

In order to minimize internal transport costs, transit ISPs try to forward packets to the closest exit point (according to Interior Gateway Protocol [IGP] cost). This behavior is known as hot-potato routing. In the distributed RR cluster model of the figure below, assume traffic coming from LA must go to Mexico. All links have the same IGP cost. If there are two exit points toward Mexico—one toward Austin and one toward Atlanta—the border router will try to send traffic to Austin based on the lower IGP cost from LA toward Austin than toward Atlanta. In a centralized RR model where the central RR resides where RR3 is (and RR1, RR2, RR4, and RR5 do not exist), the closest exit point toward Mexico, as seen from RR3, might be Atlanta. Sending the traffic from LA toward Atlanta results in suboptimal hot-potato routing, which is not desirable.

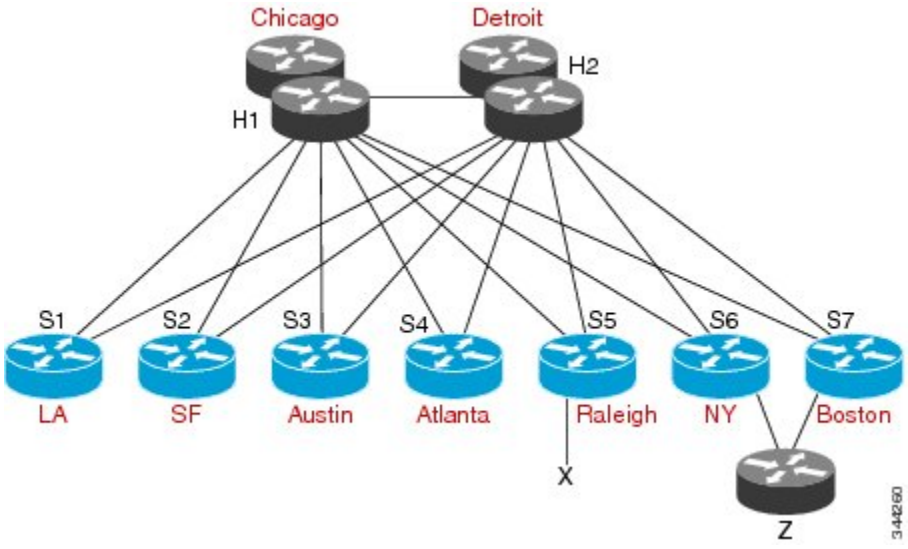
Figure 118: Distributed RR Cluster



DMVPN Scenario

In Dynamic Multipoint Virtual Private Network (DMVPN) deployments, BGP is being used for scaling. In the figure below, Z is connected to both spokes S6 (NY) and S7 (Boston). The S7 links to the hubs have lower IGP costs than the S6 links to the hubs. There are physical links not shown that connect S5 to S6 and S6 to S7, with IGP costs lower than those to the hubs. Spokes S6 and S7 will send an update to both hubs H1 (Chicago) and H2 (Detroit). The RR hubs will then select the best path based on their lower IGP cost, which might be S7. The spoke S5 (Raleigh) will receive two updates from the RRs for Z with S7 being the next hop, even though, in this scenario, it might be preferable to pick S6 (NY) as the next hop.

Figure 119: DMVPN Deployment



Benefits of BGP Additional Paths

BGP routers and route reflectors (RR) propagate only their best path over their sessions. The advertisement of a prefix replaces the previous announcement of that prefix (this in known as an implicit withdraw).

While this behavior may achieve better scaling, it can prevent path diversity, which tends to be poor or completely lost. The behavior in turn prevents efficient use of BGP multipath, prevents hitless planned maintenance, and can lead to multi-exit discriminator (MED) oscillations and suboptimal hot-potato routing. It also inhibits fast and local recovery upon nexthop failures, because the network has to wait for BGP control plane convergence to restore traffic.

The BGP Additional Paths feature is a BGP extension that allows the advertisement of multiple paths for the same prefix without the new paths implicitly replacing any previous paths. This behavior promotes path diversity and reduces MED oscillations.

BGP Additional Paths Functionality

The BGP Additional Paths feature is implemented by adding a path identifier to each path in the NLRI. The path identifier (ID) can be considered as something similar to a route distinguisher (RD) in VPNs, except that a path ID can apply to any address family. Path IDs are unique to a peering session and are generated for each network. The path identifier is used to prevent a route announcement from implicitly withdrawing the previous one. The Additional Paths feature allows the advertisement of more paths, in addition to the bestpath. The Additional Paths feature allows the advertisement of multiple paths for the same prefix, without the new paths implicitly replacing any previous paths.

The BGP Additional Paths feature requires the user to take three general steps:

1. Specify whether the device can send, receive, or send and receive additional paths. This is done at the address family level or the neighbor level, and is controlled by either the **bgp additional-paths** {**send** [**receive**] | **receive**} command or the **neighbor additional-paths** {**send** [**receive**] | **receive**} command, respectively. During session establishment, two BGP neighbors negotiate the Additional Path capabilities (whether they can send and/or receive) between them.

2. Select a set or sets of candidate paths for advertisement by specifying selection criteria (using the **bgp additional-paths select** command).

3. Advertise for a neighbor a set or sets of additional paths from the candidate paths marked (using the **neighbor advertise additional-paths** command).

To send or receive additional paths, the Additional Path capability must be negotiated. If it isn't negotiated, even if the selection criteria are such that more than the bestpath is marked and the neighbor is configured to advertise the marked paths, the selections would be useless because without the capability negotiated, only the bestpath can be sent.

Configuring BGP to send or receive additional paths triggers negotiation of additional path capability with the device's peers. Neighbors that have negotiated the capability will be grouped together in an update group (if other update group policies allow), and in a separate update group from those peers that have not negotiated the capability. Therefore, additional path capability causes the neighbor's update group membership to be recalculated.

Additional Path Selection

There are three path selection (path marking) policies, and they are not mutually exclusive. They are specified per address family, using the **bgp additional-paths select** command. They are:

- **best 2** or **best 3** (**best 2** means the bestpath and 2nd best path; the 2nd best path is the one computed by eliminating best-path from the best-computation algorithm. Similarly, **best 3** means the bestpath, 2nd best path, and 3rd best path; the 3rd best path is the one computed by eliminating bestpath and 2nd best path from the best-computation algorithm.)
- **group-best** (calculates the group-best for prefixes during bestpath calculation; described further below)
- **all** (all paths with unique next hops are eligible for selection)

Definition of the group-best Selection

The **group-best** keyword is part of the following commands:

- **advertise additional-paths**
- **bgp additional-paths select**
- **match additional-paths advertise-set**
- **neighbor advertise additional-paths**

The **group-best** is the set of paths that are the best paths from the paths of the same AS. For example, suppose there are three autonomous systems: AS 100, 200, and 300. Paths p101, p102, and p103 are from AS 100; p201, p202, and p203 are from AS200; and p301, p302, and p303 are from AS300. If we run the BGP bestpath algorithm on the paths from each AS, the algorithm will select one bestpath from each set of paths from that AS. Assuming p101 is the best from AS100, p201 is the best from AS200, and p301 is the best from AS300, then the **group-best** is the set of p101, p201, and p301.

Advertise a Subset of the Paths Selected

Take care when you select a set of paths but want to advertise a different set of paths. If the set of paths you want to advertise is not a subset of the selected paths, then you will not advertise the paths you want advertised.

The following example configures the additional paths selected to be the group-best and all selections. However, the paths configured to be advertised to the neighbor are the best 3 paths. Because the selection and advertise policy are not the same, the subsequent message is displayed. In these cases, only the bestpath is advertised.

```
Device(config)# router bgp 100
Device(config-router)# address-family ipv4
Device(config-router-af)# bgp additional-paths send receive
Device(config-router-af)# bgp additional-paths select group-best all
Device(config-router-af)# neighbor 192.168.2.2 advertise additional-paths best 3
 % BGP: AF level 'bgp additional-paths select' more restrictive than advertising policy.
This is a reminder that AF level additional-path select commands are needed.
```

How to Configure BGP Additional Paths

Configuring Additional Paths per Address Family

To select which paths are candidates to be additional paths, you can perform any combination of Steps 6, 7, and 8, as long as you perform at least one of those steps.

If you want to disable additional paths per neighbor, see the “Disabling Additional Paths per Neighbor” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**unicast** | **multicast**]
5. **bgp additional-paths** {**send** [**receive**] | **receive**}
6. **bgp additional-paths select group-best**
7. **bgp additional-paths select best** *number*
8. **bgp additional-paths select all**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name* } **advertise additional-paths** [**best** *number*] [**group-best**] [**all**]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**unicast**	**multicast**] **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode. • The following address families are supported: IPv4 unicast, IPv4 multicast, IPv4 unicast + label, IPv6 unicast, IPv6 multicast, and IPv6 multicast + label.
Step 5	**bgp additional-paths** {**send** [**receive**]	**receive**} **Example:** `Device(config-router-af)# bgp additional-paths send receive`	Enables BGP additional paths to be sent only, received only, or sent and received, after negotiation with the neighbor is completed. • This example enables additional paths to be sent and received.

	Command or Action	Purpose
Step 6	**bgp additional-paths select group-best** **Example:** `Device(config-router-af)# bgp additional-paths select group-best`	(Optional) Calculates the group-best for prefixes during bestpath calculation.
Step 7	**bgp additional-paths select best** *number* **Example:** `Device(config-router-af)# bgp additional-paths select best 3`	(Optional) Calculates the specified number of best paths, including the advertisement of the bestpath. • The value of *number* can be 2 or 3.
Step 8	**bgp additional-paths select all** **Example:** `Device(config-router-af)# bgp additional-paths select all`	(Optional) Specifies that all paths with unique next hops are eligible for selection.
Step 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name* } **advertise additional-paths** [**best** *number*] [**group-best**] [**all**] **Example:** `Device(config-router-af)# neighbor 192.168.0.1 advertise additional-paths best 3 group-best all`	Specifies which selection methods control the additional paths that are advertised to the neighbor.
Step 10	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring Additional Paths per Neighbor

To select which paths are candidates to be additional paths, you can perform any combination of Steps 6, 7, and 8, as long as you perform at least one of those steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**unicast** | **multicast**]
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **additional-paths** {**send** [**receive**] | **receive**}
6. **bgp additional-paths select group-best**
7. **bgp additional-paths select best** *number*
8. **bgp additional-paths select all**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **advertise additional-paths** [**best** *number*] [**group-best**] [**all**]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters address family configuration mode. • The following address families are supported: IPv4 unicast, IPv4 multicast, IPv4 unicast + label, IPv6 unicast, IPv6 multicast, and IPv6 unicast+ label.
Step 5	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **additional-paths** {**send** [**receive**] | **receive**} **Example:** `Device(config-router-af)# neighbor 192.168.1.2 additional-paths send receive`	Enables the neighbor to send or receive additional paths after negotiation is completed. • This example enables the neighbor to send and receive additional paths. • Note that this command overrides any send or receive capability that might have been configured at the address-family level.
Step 6	**bgp additional-paths select group-best** **Example:** `Device(config-router-af)# bgp additional-paths select group-best`	(Optional) Calculates the group-best for prefixes during bestpath calculation.
Step 7	**bgp additional-paths select best** *number* **Example:** `Device(config-router-af)# bgp additional-paths select best 3`	(Optional) Calculates the specified number of best paths, including the selection of the bestpath. • The value of *number* can be 2 or 3.
Step 8	**bgp additional-paths select all** **Example:**	(Optional) Specifies that all paths with unique next hops are eligible for selection.

	Command or Action	Purpose
	`Device(config-router-af)# bgp additional-paths select all`	
Step 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **advertise additional-paths** [**best** *number*] [**group-best**] [**all**] **Example:** `Device(config-router-af)# neighbor 192.168.1.2 advertise additional-paths best 3 group-best all`	Specifies the selection methods that control which additional paths are advertised for the neighbor.
Step 10	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuring Additional Paths Using a Peer Policy Template

In this configuration task example, the capability to send and receive additional paths and the selection criteria are configured for the address family, and then the template is configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 unicast**
5. **bgp additional-paths** {**send** [**receive**] | **receive**}
6. **bgp additional-paths select** [**best** *number*] [**group-best**] [**all**]
7. **template peer-policy** *policy-template-name*
8. **additional-paths** {**send** [**receive**] | **receive**}
9. **advertise additional-paths** [**best** *number*] [**group-best**] [**all**]
10. **exit**
11. **address-family ipv4 unicast**
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
13. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4 unicast** **Example:** `Device(config-router)# address-family ipv4 unicast`	Configures the IPv4 address family.
Step 5	**bgp additional-paths** {**send** [**receive**] | **receive**} **Example:** `Device(config-router)# bgp additional-paths send receive`	Enables BGP additional paths to be sent only, received only, or sent and received for the peers in the address family.
Step 6	**bgp additional-paths select** [**best** *number*] [**group-best**] [**all**] **Example:** `Device(config-router)# bgp additional-paths select best 3 group-best all`	Causes the system to calculate BGP additional paths that can be candidates for advertisement in addition to a bestpath.
Step 7	**template peer-policy** *policy-template-name* **Example:** `Device(config-router)# template peer-policy rr-client-pt1`	Enters policy-template configuration mode and creates a peer policy template.
Step 8	**additional-paths** {**send** [**receive**] | **receive**} **Example:** `Device(config-router-ptmp)# additional-paths send receive`	Enables BGP additional paths to be sent only, received only, or sent and received for the peers covered by the peer policy template.
Step 9	**advertise additional-paths** [**best** *number*] [**group-best**] [**all**] **Example:** `Device(config-router-ptmp)# advertise additional-paths best 3 group-best all`	Specifies the selection methods that control which additional paths are advertised for the peers covered by the peer policy template.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config-router-ptmp)# exit`	Exits policy-template configuration mode and returns to router configuration mode.
Step 11	**address-family ipv4 unicast** **Example:** `Device(config-router)# address-family ipv4 unicast`	Configures the IPv4 address family.
Step 12	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 remote-as 45000`	Adds an entry to the BGP neighbor table.
Step 13	**neighbor** *ip-address* **inherit peer-policy** *policy-template-name* **Example:** `Device(config-router-af)# neighbor 192.168.1.1 inherit peer-policy rr-client-pt1`	Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.
Step 14	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Filtering and Setting Actions for Additional Paths

You can optionally use a route map to filter the paths to be advertised by matching on the tags of additional paths that are candidates to be advertised. (These tags are the advertise-sets that are configured with the **bgp additional-paths select** command.) Paths that have the same path marking (tag) as the marking that is configured in the **match additional-paths advertise-set** command match the route map entry (and are permitted or denied).

You can also optionally set one or more actions to take for those paths that pass the route map. This task happens to use the **set metric** command to illustrate using a route map with the **match additional-paths advertise-set** command. Of course, other **set** commands are available that are not shown in this task.

Why set a metric for paths marked with **all** (all paths with a unique next hop)? Suppose the neighbor 2001:DB8::1037 is receiving the same route from different neighbors. Routes received from the local device have a metric of 565 and routes from another device perhaps have a metric of 700. Routes with metric 565 will have precedence over the routes with metric 700.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match additional-paths advertise-set** [**best** *number*] [**best-range** *start-range end-range*] [**group-best**] [**all**]
5. **set metric** *metric-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] **Example:** `Device(config)# route-map additional_path1 permit 10`	Creates a route map.
Step 4	**match additional-paths advertise-set** [**best** *number*] [**best-range** *start-range end-range*] [**group-best**] [**all**] **Example:** `Device(config-route-map)# match additional-paths advertise-set best 3`	Matches on any path that is tagged with the specified path selection policy. • You must specify at least one selection method; you can specify more than one selection method in the command. • Specifying **best** *number* is incompatible with specifying **best-range**. • Specifying **best 1** will match only the bestpath. • Specifying **best-range 1 1** will match only the bestpath. • Only one **match additional-paths advertise-set** command is allowed per route map. A subsequent **match additional-paths advertise-set** command will overwrite the previous command.
Step 5	**set metric** *metric-value* **Example:** `Device(config-route-map)# set metric 500`	Sets the metric of the additional paths that pass the match criteria. • Note that other **set** commands can be used to take action on the paths that pass the route map. This example happens to use the **set metric** command.

What to do next

After creating the route map, you would reference the route map in the **neighbor route-map out** command. Thus, the route map is applied to paths being advertised (outgoing) to neighbors. Then you would use the **neighbor advertise additional-paths** command to advertise the additional paths. See the "Example: BGP Additional Paths" section to see the route map in context.

Displaying Additional Path Information

Perform either Step 2 or Step 3 in this task to see information about BGP additional paths.

SUMMARY STEPS

1. **enable**
2. **show ip bgp neighbors** [*ip-address*]
3. **show ip bgp** [*network*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp neighbors** [*ip-address*] **Example:** `Device# show ip bgp neighbors 192.168.1.1`	Displays the capabilities of the neighbor to send and receive additional paths.
Step 3	**show ip bgp** [*network*] **Example:** `Device# show ip bgp 192.168.0.0`	Displays the additional path selections and path ID for the network.

Disabling Additional Paths per Neighbor

If you had configured the sending or receiving of additional paths on a per neighbor basis (with the **neighbor additional-paths** command), and you wanted to disable that functionality, you would use the **no neighbor additional-paths** command.

However, if you had configured the sending or receiving of additional paths for an address family (with the **bgp additional-paths** command), and you wanted to disable that functionality for a neighbor, you would use the **neighbor additional-paths disable** command. Disabling additional paths also works if the functionality was inherited from a template.

Perform this task to disable additional path capability for a neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **bgp additional-paths** {**send** [**receive**] | **receive**}
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **additional-paths disable**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv6** [**unicast**	**multicast**] **Example:** `Device(config-router)# address-family ipv6 unicast`	Enters address family configuration mode.
Step 5	**bgp additional-paths** {**send** [**receive**]	**receive**} **Example:** `Device(config-router-af)# bgp additional-paths send receive`	Enables BGP additional paths to be sent or received for the neighbors in the address family.
Step 6	**neighbor** {*ip-address*	*ipv6-address*	*peer-group-name*} **additional-paths disable** **Example:** `Device(config-router-af)# neighbor 2001:DB8::1 additional-paths disable`	Disables BGP additional paths from being sent to or received from the specified neighbor. • The additional path functionality is still enabled for the rest of the neighbors in the address family.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Configuration Examples for BGP Additional Paths

Example: BGP Additional Path Send and Receive Capabilities

In this example, R1's address is 192.168.1.1; its neighbor is R2, which has address 192.168.1.2. Updates are sent from R2 to R1 with additional-paths (all paths advertised). Updates are sent from R1 to R2 with only the classic BGP bestpath advertised because R2 is only able to send additional paths, not receive additional paths.

R1

```
router bgp 1
 address-family ipv4 unicast
 bgp additional-paths select all
 neighbor 192.168.1.2 additional-paths send receive
 neighbor 192.168.1.2 advertise additional-paths all
```

R2

```
router bgp 2
 address-family ipv4 unicast
 bgp additional-paths select all
 neighbor 192.168.1.1 additional-paths send
 neighbor 192.168.1.1 advertise additional-paths all
```

Example: BGP Additional Paths

In the following example, for every address family, there are one or more eBGP neighbors not shown in the configuration that are sending routes to the local device. The eBGP routes learned from those neighbors are advertised toward the neighbors shown in the configuration below and the path attributes are changed. The example configures that:

- The route map called add_path1 specifies that all the paths are advertised toward neighbor 192.168.101.15, but any path that is marked with **best 2** will have its metric set to 780 before being sent toward that neighbor.
- The route map called add_path2 specifies that any path that is marked with **best 3** will have its metric set to 640 and will be advertised toward neighbor 192.168.25.
- The route map called add_path3 specifies that any path that is marked with **group-best** will have its metric set to 825 and will be advertised toward neighbor 2001:DB8::1045.
- In the IPv6 multicast address family, all paths are candidates to be advertised and will be advertised toward neighbor 2001:DB8::1037.

```
router bgp 1
 neighbor 192.168.101.15 remote-as 1
 neighbor 192.168.101.25 remote-as 1
 neighbor 2001:DB8::1045 remote-as 1
 neighbor 2001:DB8::1037 remote-as 1
!
 address-family ipv4 unicast
  bgp additional-paths send receive
  bgp additional-paths select all best 3 group-best
  neighbor 192.168.101.15 activate
  neighbor 192.168.101.15 route-map add_path1 out
  neighbor 192.168.101.15 advertise additional-paths best 2
 exit-address-family
!
address-family ipv4 multicast
  bgp additional-paths send receive
  bgp additional-paths select all best 3 group-best
  neighbor 192.168.101.25 activate
  neighbor 192.168.101.25 route-map add_path2 out
  neighbor 192.168.101.25 advertise additional-paths best 3
 exit-address-family
!
 address-family ipv6 unicast
  bgp additional-paths send receive
  bgp additional-paths select group-best
  neighbor 2001:DB8::1045 activate
  neighbor 2001:DB8::1045 route-map add_path3 out
  neighbor 2001:DB8::1045 advertise additional-paths all group-best
 exit-address-family
!
 address-family ipv6 multicast
  bgp additional-paths send receive
  bgp additional-paths select all
  neighbor 2001:DB8::1037 activate
  neighbor 2001:DB8::1037 route-map add_path4 out
  neighbor 2001:DB8::1037 advertise additional-paths all
 exit-address-family
!
route-map add_path1 permit 10
 match additional-paths advertise-set best 2
 set metric 780
route-map add_path1 permit 20
!
route-map add_path2 permit 10
 match additional-paths advertise-set best 3
 set metric 640
!
route-map add_path3 permit 10
 match additional-paths advertise-set group-best
 set metric 825
!
```

Example: Neighbor Capabilities Override Address Family Capabilities

In the following example, the receive-only capability of the neighbor overrides the send and receive capability of the address family:

```
router bgp 65000
```

```
 address-family ipv6 multicast
 bgp additional-paths send receive
 bgp additional-paths select group-best
 neighbor 2001:DB8::1037 activate
 neighbor 2001:DB8::1037 additional-paths receive
 neighbor 2001:DB8::1037 advertise additional-paths group-best
!
```

Example: BGP Additional Paths Using a Peer Policy Template

```
router bgp 45000
address-family ipv4 unicast
 bgp additional-paths send receive
 bgp additional-paths select all group-best best 3
 template peer-policy rr-client-pt1
  additional-paths send receive
  advertise additional-paths group-best best 3
  exit
 address-family ipv4 unicast
  neighbor 192.168.1.1 remote-as 45000
  neighbor 192.168.1.1 inherit peer-policy rr-client-pt1
  end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 3107	*Carrying Label Information in BGP-4*
RFC 4271	*A Border Gateway Protocl (BGP-4)*
RFC 4760	*Multiprotocol Extensions for BGP-4*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Additional Paths

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 133: Feature Information for BGP Additional Paths

Feature Name	Releases	Feature Information
BGP Additional Paths		The BGP Additional Paths feature allows the advertisement of multiple paths for the same prefix without the new paths implicitly replacing any previous paths. The following commands were introduced: • **additional-paths** • **advertise additional-paths** • **bgp additional-paths** • **bgp additional-paths select** • **match additional-paths advertise-set** • **neighbor additional-paths** • **neighbor advertise additional-paths** The following commands were modified: • **show ip bgp** • **show ip bgp neighbors**

CHAPTER 104

BGP-Multiple Cluster IDs

The BGP—Multiple Cluster IDs feature allows an iBGP neighbor (usually a route reflector) to have multiple cluster IDs: a global cluster ID and additional cluster IDs that are assigned to clients (neighbors). Prior to the introduction of this feature, a device could have a single, global cluster ID.

When a network administrator configures per-neighbor cluster IDs:

- The loop prevention mechanism based on a CLUSTER_LIST is automatically modified to take into account multiple cluster IDs.
- A network administrator can disable client-to-client route reflection based on cluster ID.

Information About BGP-Multiple Cluster IDs

Benefit of Multiple Cluster IDs Per Route Reflector

The BGP—Multiple Cluster IDs feature allows a route reflector (RR) to belong to multiple clusters, and therefore have multiple cluster IDs. An RR can have a cluster ID configured on a global basis and a per-neighbor basis. A single cluster ID can be assigned to two or more iBGP neighbors. Prior to this feature, an RR had a single, global cluster ID, which was configured by the **bgp cluster-id** router configuration command.

When a cluster ID is configured per neighbor (by the **neighbor cluster-id** router configuration command), the following two changes occur:

- The loop prevention mechanism based on the CLUSTER_LIST attribute is automatically modified to take into account multiple cluster IDs.
- The network administrator can disable client-to-client route reflection based on cluster ID, which allows the network design to change.

The loop prevention mechanism and the CLUSTER_LIST propagation rules are described in the section "How a CLUSTER_LIST Attribute is Used." Disabling client-to-client reflection is described in the section "Behaviors When Disabling Client-to-Client Route Reflection."

How a CLUSTER_LIST Attribute is Used

The CLUSTER_LIST propagation rules differ among releases, depending on whether the device is running a Cisco software release generated before or after the BGP—Multiple Cluster IDs feature was implemented. The same is true for loop prevention based on the CLUSTER_LIST.

The CLUSTER_LIST behavior is described below. Classic refers to the behavior of software released before the multiple cluster IDs feature was implemented; MCID refers to the behavior of software released after the feature was implemented.

CLUSTER_LIST Propagation Rules

- Classic—Before reflecting a route, the RR appends the global cluster ID to the CLUSTER_LIST. If the received route had no CLUSTER_LIST attribute, the RR creates a new CLUSTER_LIST attribute with that global cluster ID.
- MCID—Before reflecting a route, the RR appends the cluster ID of the neighbor the route was received from to the CLUSTER_LIST. If the received route had no CLUSTER_LIST attribute, the RR creates a new CLUSTER_LIST attribute with that cluster ID. This behavior includes a neighbor that is not a client of the speaker. If the nonclient neighbor the route was received from does not have an associated cluster ID, the RR uses the global cluster ID.

Loop Prevention Based on CLUSTER_LIST

- Classic—When receiving a route, the RR discards the route if the RR's global cluster ID is contained in the CLUSTER_LIST of the route.
- MCID—When receiving a route, the RR discards the route if the RR's global cluster ID or any of the cluster IDs assigned to any of the iBGP neighbors is contained in the CLUSTER_LIST of the route.

Behaviors When Disabling Client-to-Client Route Reflection

With the introduction of multiple cluster IDs per iBGP neighbor, it is possible to disable route reflection from client to client on the basis of cluster ID. Disabling route reflection allows you to change the network design. A typical (but not required) scenario after disabling route reflection is that clients are fully meshed, so they have to send more updates, and the RR has client-to-client reflection disabled, so that it has to send fewer updates.

You might want to disable route reflection in a scenario similar to the one in the figure below. An RR has several clients [Provider-Edge (PE) routers] with which it has sessions. The iBGP neighbors that should belong to one cluster were assigned the same cluster ID.

Because the PEs belonging to the same cluster are fully meshed (PE1 and PE2 have a session between them; PE3 and PE4 have a session between them), there is no need to reflect the routes between them. That is, routes from PE1 should be forwarded to PE3 and PE4, but not to PE2.

It is important to know that when the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

Disabling client-to-client route reflection is done differently and has different results, depending on whether the device is running Cisco software generated before or after the multiple cluster IDs feature was implemented. Classic refers to the behavior of software released before the multiple cluster IDs feature was implemented; MCID refers to the behavior of software released after the multiple cluster IDs feature was implemented.

- Classic—When receiving a route from a client, the RR does not reflect it to any other client. Other scenarios for reflection (client-to-nonclient and nonclient-to-client) are maintained. Disabling of route reflection from client to client is usually done when all the clients are fully meshed (the routes are advertised between the clients via that mesh, so there is no need for reflection). The command to disable client-to-client route reflection is entered in router configuration mode (after the **router bgp** command) and it applies globally to all address families: **no bgp client-to-client reflection**
- MCID—When receiving a route from a client, the RR does not reflect it to another client if both clients belong to a cluster for which client-to-client reflection has been disabled. Therefore, route reflection is disabled only intracluster (within the cluster specified). Other cases for reflection (client-to-nonclient, nonclient-to-client, and intercluster) are maintained. This functionality is usually configured when all the clients for a particular cluster are fully meshed among themselves (but not with clients of other clusters). The command to disable client-to-client route reflection for a particular cluster is entered in router configuration mode and it applies globally to all address families:

  no bgp client-to-client reflection intra-cluster cluster-id {any | *cluster-id1 cluster-id2...***}**

  The **any** keyword is used to disable client-to-client reflection for any cluster.

  The Classic, previously released command for disabling all client-to-client reflection is also still available during this post-MCID release timeframe:

  no bgp client-to-client reflection [all]

  (The optional **all** keyword has no effect in either the positive or negative form of the command, and does not appear in configuration files. It is just to remind the network administrator that both intercluster and intracluster client-to-client reflection are enabled or disabled.)

  In summary, after the introduction of the multiple cluster IDs feature, there are three levels of configuration that can disable client-to-client reflection. The software performs them in the following order, from least specific to most specific:

  1. Least specific: **no bgp client-to-client reflection [all]** Disables intracluster and intercluster client-to-client reflection.
  2. More specific: **no bgp client-to-client reflection intra-cluster cluster-id any** Disables intracluster client-to-client reflection for any cluster-id.
  3. Most specific: **no bgp client-to-client reflection intra-cluster cluster-id** *cluster-id1 cluster-id2 ...* Disables intracluster client-to-client reflection for the specified clusters.

  When BGP is advertising updates, the software evaluates each level of configuration in order. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is necessary.

  Note the results of the base (positive) and negative (**no**) forms of the three commands listed above:

  - A negative configuration (that is, with the **no** keyword) overwrites any less specific configuration.
  - A positive configuration (that is, without the **no** keyword) will lose out to (default to) what is configured in a less specific configuration.
  - Configurations at any level appear in the configuration file only if they are negative.

All levels can be configured independently and all levels appear in the configuration file independently of the configuration of other levels.

Note that negative configuration makes any more specific configuration unnecessary (because even if the more specific configuration is positive, it is not processed after the negative configuration; if the more specific configuration is negative, it is functionally the same as the earlier negative configuration). The following examples illustrate this behavior.

Example 1

no bgp client-to-client reflection

no bgp client-to-client reflection intra-cluster cluster-id any

Intercluster and intracluster reflection are disabled (based on the first command). The second command disables intracluster reflection, but it is unnecessary because intracluster reflection is already disabled by the first command.

Example 2

no bgp client-to-client reflection intra-cluster cluster-id any

bgp client-to-client reflection intra-cluster cluster-id 1.1.1.1

Cluster ID 1.1.1.1 has intracluster route reflection disabled (even though the second command is positive), because the first command is used to evaluate the update. The first command was negative, and once any level of configuration disables client-to-client reflection, no further evaluation is performed.

Another way to look at this example is that the second command, because it is in a positive form, defaults to the behavior of the first command (which is less specific). Thus, the second command is unnecessary.

Note that the second command would not appear in a configuration file because it is not a negative command.

How to Use BGP-Multiple Cluster IDs

Configuring a Cluster ID per Neighbor

Perform this task on an iBGP peer (usually a route reflector) to configure a cluster ID per neighbor. Configuring a cluster ID per neighbor causes the loop-prevention mechanism based on the CLUSTER_LIST to be automatically modified to take into account multiple cluster IDs. Also, you gain the ability to disable client-to-client route reflection on the basis of cluster ID. The software tags the neighbor so that you can disable route reflection with the use of another command. (See the tasks for disabling client-to-client reflection later in this module.)

Note When you change a cluster ID for a neighbor, BGP automatically does an inbound soft refresh and an outbound soft refresh for all iBGP peers.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *autonomous-system-number*
5. **neighbor** {*ip-address* | *ipv6-address*} **cluster-id** *cluster-id*
6. **end**
7. **show ip bgp cluster-ids**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *ipv6-address*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.1.2 remote-as 65000`	Adds an entry to the BGP routing table.
Step 5	**neighbor** {*ip-address* | *ipv6-address*} **cluster-id** *cluster-id* **Example:** `Device(config-router)# neighbor 192.168.1.2 cluster-id 0.0.0.1`	Assigns a cluster ID to the specified neighbor. • The cluster ID can be in dotted decimal format (such as 192.168.7.4) or decimal format (such as 23), with a maximum of 4 bytes. • A cluster ID that is configured in decimal format (such as 23) is modified to dotted decimal format (such as 0.0.0.23) when it appears in a configuration file. • When you change a cluster ID for a neighbor, BGP automatically does an inbound soft refresh and an outbound soft refresh for all iBGP peers.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-router)# end`	(Optional) Exits to privileged EXEC mode.
Step 7	**show ip bgp cluster-ids** **Example:** `Device# show ip bgp cluster-ids`	(Optional) Lists: • the global cluster ID (whether configured or not) • all cluster IDs that are configured to a neighbor • all cluster IDs for which the network administrator has disabled reflection

Disabling Intracluster and Intercluster Client-to-Client Reflection

Perform the following task on a route reflector if you want to disable both intracluster and intercluster client-to-client reflection. Doing so is the broadest (least specific) way to disable client-to-client reflection. Before advertising updates, the software evaluates each level of configuration in order from least specific to most specific. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is needed.

Note When the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp client-to-client reflection** [**all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp client-to-client reflection** [**all**] **Example:** `Device(config-router)# no bgp client-to-client reflection all`	Disables intracluster and intercluster client-to-client route reflection. • The **all** keyword is just to emphasize that the **bgp client-to-client reflection** command affects both intracluster and intercluster reflection; the **all** keyword has no effect in the positive or negative form of the command.

Disabling Intracluster Client-to-Client Reflection for Any Cluster ID

Perform the following task on a route reflector to disable intracluster client-to-client reflection for any cluster ID. Doing so is considered to be the middle of the three levels of commands available to disable client-to-client reflection. That is, it is more specific than disabling intracluster and intercluster client-to-client reflection, but it is not as specific as disabling intracluster client-to-client reflection for certain cluster IDs.

Before advertising updates, the software evaluates each level of configuration in order from least specific to most specific. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is needed.

Note When the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp client-to-client reflection intra-cluster cluster-id any**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp client-to-client reflection intra-cluster cluster-id any** **Example:** `Device(config-router)# no bgp client-to-client reflection intra-cluster cluster-id any`	Disables intracluster client-to-client route reflection for any cluster.

Disabling Intracluster Client-to-Client Reflection for Specified Cluster IDs

Perform the following task on a route reflector to disable intracluster client-to-client reflection for specified cluster IDs. Doing so is considered to be the most specific of the three levels of commands available to disable client-to-client reflection. Before advertising updates, the software evaluates each level of configuration in order from least specific to most specific. Once any level of configuration disables client-to-client reflection, no further evaluation of more specific policies is needed.

Note When the software changes reflection state for a given cluster ID, BGP sends an outbound soft refresh to all clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp client-to-client reflection intra-cluster cluster-id** *cluster-id1* [*cluster-id2...*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp client-to-client reflection intra-cluster cluster-id** *cluster-id1* [*cluster-id2...*] **Example:** `Device(config-router)# no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1 0.0.0.3 105`	Disables intracluster client-to-client route reflection within each of the specified clusters. • Note that this example command will appear in the configuration file as "no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1 0.0.0.3 0.0.0.105" because decimal cluster ID numbers appear in the dotted decimal format.

Configuration Examples for BGP-Multiple Cluster IDs

Example: Per-Neighbor Cluster ID

The following example is configured on a route reflector. The neighbor (client) at IPv6 address 2001:DB8:1::1 is configured to have the cluster ID of 0.0.0.6:

```
router bgp 6500
 neighbor 2001:DB8:1::1 cluster-id 0.0.0.6
```

Example: Disabling Client-to-Client Reflection

The following example disables all intracluster and intercluster client-to-client reflection:

```
router bgp 65000
 no bgp client-to-client reflection all
```

The following example disables intracluster client-to-client reflection for any cluster ID:

```
router bgp 65000
 no bgp client-to-client reflection intra-cluster cluster-id any
```

The following example disables intracluster client-to-client reflection for the specified cluster IDs 0.0.0.1, 14, 15, and 0.0.0.6:

```
router bgp 65000
 no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1 14 15 0.0.0.6
```

Remember that a cluster ID specified in the **neighbor cluster-id** command in decimal format (such as 23) will appear in a configuration file in dotted decimal format (such as 0.0.0.23). The decimal format does not appear in the configuration file. The running configuration might look like this:

```
router bgp 65000
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.1
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.6
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.14
no bgp client-to-client reflection intra-cluster cluster-id 0.0.0.15
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP-Multiple Cluster IDs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 134: Feature Information for BGP—Multiple Cluster IDs

Feature Name	Releases	Feature Information
BGP—Multiple Cluster IDs		The BGP—Multiple Cluster IDs feature allows an iBGP neighbor (usually a route reflector) to have multiple cluster IDs: a global cluster ID and additional cluster IDs that are assigned to clients (neighbors). Prior to the introduction of this feature, a device could have a single, global cluster ID. When a network administrator configures per-neighbor cluster IDs: • The loop prevention mechanism based on a CLUSTER_LIST is automatically modified to take into account multiple cluster IDs. • A network administrator can disable client-to-client route reflection based on cluster ID. The following commands were introduced: • **bgp client-to-client reflection intra-cluster** • **neighbor cluster-id** • **show ip bgp cluster-ids** The following commands were modified: • **bgp client-to-client reflection** • **show ip bgp neighbors** • **show ip bgp template peer-session** • **show ip bgp update-group**

CHAPTER 105

BGP-VPN Distinguisher Attribute

The BGP—VPN Distinguisher Attribute feature allows a network administrator to keep source route targets (RTs) private from an Autonomous System Border Router (ASBR) in a destination autonomous system. An RT at an egress ASBR is mapped to a VPN distinguisher, the VPN distinguisher is carried through the eBGP, and then it is mapped to an RT at the ingress ASBR.

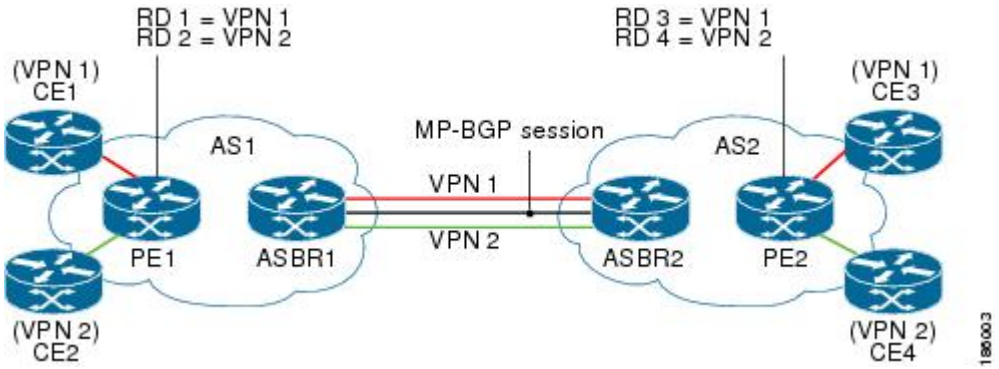
Information About BGP-VPN Distinguisher Attribute

Role and Benefit of the VPN Distinguisher Attribute

Route-target (RT) extended community attributes identify the VPN membership of routes. The RT attributes are placed onto a route at the exporting (egress) provider edge router (PE) and are transported across the iBGP cloud and across autonomous systems. Any Virtual Routing and Forwarding (VRF) instances at the remote PE that want to import such routes must have the corresponding RTs set as import RTs for that VRF.

The figure below illustrates two autonomous systems, each containing customer edge routers (CEs) that belong to different VPNs. Each PE tracks which route distinguisher (RD) corresponds to which VPN, thus controlling the traffic that belongs to each VPN.

Figure 121: Scenario in Which ASBRs Translate RTs Between Autonomous Systems



In an Inter-AS Option B scenario like the one in the figure above, these routes are carried across an AS boundary from Autonomous System Border Router 1 (ASBR1) to ASBR2 over an MP-eBGP session, with the routes' respective RTs as extended community attributes being received by ASBR2.

ASBR2 must maintain complex RT mapping schemes to translate RTs originated by AS1 to RTs recognized by AS2, so that the RTs can be imported by their respective VPN membership CE connections on PE2 for CE3 and CE4.

Some network administrators prefer to hide the RTs they source in AS1 from devices in AS2. In order to do that, the administrator must differentiate routes belonging to each VPN with a certain attribute so that the RTs can be removed on the outbound side of ASBR1 before sending routes to ASBR2, and ASBR2 can then map that attribute to recognizable RTs in AS2. The VPN Distinguisher (VD) extended community attribute serves that purpose.

The benefit of the BGP—VPN Distinguisher Attribute feature is that source RTs can be kept private from devices in destination autonomous systems.

How the VPN Distinguisher Attribute Works

The network administrator configures the egress ASBR to perform translation of RTs to a VPN distinguisher extended community attribute, and configures the ingress ASBR to perform translation of the VPN distinguisher to RTs. More specifically, the translation is achieved as follows:

On the Egress ASBR

- An outbound route map specifies a **match excommunity** clause that determines which VPN routes are subject to mapping, based on the route's RT values.
- A **set extcommunity vpn-distinguisher** command sets the VPN distinguisher that replaces the RTs.
- The **set extcomm-list delete** command that references the same set of RTs is configured to remove the RTs, and then the route is sent to the neighboring ingress ASBR.

On the Ingress ARBR

- An inbound route map specifies a **match excommunity vpn-distinguisher** command that determines which VPN routes are subject to mapping, based on the route's VPN distinguisher.
- The **set extcommunity rt** command specifies the RTs that replace the VPN distinguisher.
- For routes that match the clause, the VPN distinguisher is replaced with the configured RTs.

Additional Behaviors Related to the VPN Distinguisher

On the egress ASBR, if a VPN route matches a route map clause that does not have the **set extcommunity vpn-distinguisher** command configured, the RTs that the VPN route is tagged with are retained.

The VPN distinguisher is transitive across the AS boundary, but is not carried within the iBGP cloud. That is, the ingress ASBR can receive the VPN distinguisher from an eBGP peer, but the VPN distinguisher is discarded on the inbound side after it is mapped to the corresponding RTs.

On the ingress ASBR, if a VPN route carrying the VPN distinguisher matches a route map clause that does not have a **set extcommunity rt** command configured in the inbound route map, the system does not discard the attribute, nor does it propagate the attribute within the iBGP cloud. The VPN distinguisher for the route is retained so that the network administrator can configure the correct inbound policy to translate the VPN distinguisher to the RTs that the VPN route should carry. If the route is sent to eBGP peers, the VPN distinguisher is carried as is. The network administrator could configure a route-map entry to remove the VPN distinguisher from routes sent to eBGP peers.

Configuring a **set extcommunity vpn-distinguisher** command in an outbound route map or a **match excommunity** command in an inbound route map results in an outbound or inbound route refresh request, respectively, in order to update the routes being sent or received.

How to Configure BGP-VPN Distinguisher Attribute

Replacing an RT with a VPN Distinguisher Attribute

Perform this task on an egress ASBR to replace a route target (RT) with a VPN distinguisher extended community attribute. Remember to replace the VPN distinguisher with a route target on the ingress ASBR; that task is described in the “Replacing a VPN Distinguisher Attribute with an RT” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *value*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**
8. **set extcommunity vpn-distinguisher** *id*
9. **exit**
10. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **route-map** *map-name* **out**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *value* **Example:** `Device(config)# ip extcommunity-list 4 permit rt 101:100`	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified RT are in the extended community list. • This example permits routes having RT 101:100 into the extended community list 4.
Step 4	**exit** **Example:** `Device(config-extcomm-list)# exit`	Exits the configuration mode and enters the next higher configuration mode.
Step 5	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map vpn-id-map1 permit 10`	Configures a route map that permits or denies the routes allowed by the subsequent **match** command. • This example permits the routes allowed by the subsequent **match** command.
Step 6	**match extcommunity** *extended-community-list-name* **Example:** `Device(config-route-map)# match extcommunity 4`	Matches on the specified community list. • For this example, routes that match the extended community list 4 (which was configured in Step 3) are subject to the subsequent **set** commands.
Step 7	**set extcomm-list** *extcommunity-name* **delete** **Example:** `Device(config-route-map)# set extcomm-list 4 delete`	Deletes the RT from routes that are in the specified extended community list. • For this example, RTs are deleted from routes that are in extended community list 4.
Step 8	**set extcommunity vpn-distinguisher** *id* **Example:** `Device(config-route-map)# set extcommunity vpn-distinguisher 111:100`	For the routes that are permitted by the route map, sets the specified VPN distinguisher. • For this example, routes that match extended community 4 have their VPN distinguisher set to 111:100.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map vpn-id-map1 permit 20`	(Optional) Configures a route map entry that permits routes. • This example configures a route map entry that permits other routes not subject to the RT-to-VPN distinguisher mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 12	**router bgp** *as-number* **Example:** `Device(config)# router bgp 2000`	Enters router configuration mode and creates a BGP routing process.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.101.1 remote-as 2000`	Specifies that the neighbor belongs to the autonomous system.
Step 14	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.101.1 activate`	Activates the specified neighbor.
Step 16	**neighbor** *ip-address* **route-map** *map-name* **out** **Example:** `Device(config-router-af)# neighbor 192.168.101.1 route-map vpn-id-map1 out`	Applies the specified outgoing route map to the specified neighbor.

	Command or Action	Purpose
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters privileged EXEC mode.

Replacing a VPN Distingusher Attribute with an RT

Perform this task on an ingress ASBR to replace a VPN distinguisher extended community attribute with a route target (RT) attribute. This task assumes you already configured the egress ASBR to replace the RT with a VPN distinguisher; that task is described in the "Replacing an RT with a VPN Distinguisher Attribute" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} **vpn-distinguisher** *id*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**
8. **set extcommunity rt** *value* **additive**
9. **exit**
10. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **route-map** *map-name* **in**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **vpn-distinguisher** *id* **Example:** `Device(config)# ip extcommunity-list 51 permit vpn-distinguisher 111:100`	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified VPN distinguisher are in the extended community list. • This example permits routes having VPN distinguisher 111:110 into the extended community list 51.
Step 4	**exit** **Example:** `Device(config-extcomm-list)# exit`	Exits the configuration mode and enters the next higher configuration mode.
Step 5	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map vpn-id-rewrite-map1 permit 10`	Configures a route map that permits or denies the routes allowed by the subsequent **match** command. • This example permits the routes allowed by the subsequent **match** command.
Step 6	**match extcommunity** *extended-community-list-name* **Example:** `Device(config-route-map)# match extcommunity 51`	Matches on the specified community list. • For this example, routes that match the extended community list 51 (which was configured in Step 3) are subject to the subsequent **set** commands.
Step 7	**set extcomm-list** *extcommunity-name* **delete** **Example:** `Device(config-route-map)# set extcomm-list 51 delete`	Deletes the VPN distinguisher from routes that are in the specified extended community list. • For this example, VPN distinguishers are deleted from routes that are in extended community list 51.
Step 8	**set extcommunity rt** *value* **additive** **Example:** `Device(config-route-map)# set extcommunity rt 101:1 additive`	Sets the routes that are permitted by the route map with the specified RT. • For this example, routes that match extended community 51 have their RT set to 101:1. The **additive** keyword causes the RT to be added to the RT list without replacing any RTs.
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:**	(Optional) Configures a route map entry that permits routes. • This example configures a route map entry that permits other routes not subject to the VPN

	Command or Action	Purpose
	`Device(config)# route-map vpn-id-rewrite-map1 permit 20`	distinguisher-to-RT mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 12	**router bgp** *as-number* **Example:** `Device(config)# router bgp 3000`	Enters router configuration mode and creates a BGP routing process.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.0.81 remote-as 3000`	Specifies that the neighbor belongs to the autonomous system.
Step 14	**address-family vpnv4** **Example:** `Device(config-router-af)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.0.81 activate`	Activates the specified neighbor.
Step 16	**neighbor** *ip-address* **route-map** *map-name* **in** **Example:** `Device(config-router-af)# neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in`	Applies the specified outgoing route map to the specified neighbor.
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters privileged EXEC mode.

Example

Configuration Examples for BGP-VPN Distinguisher Attribute

Example: Translating RT to VPN Distinguisher to RT

The following example shows the egress ASBR configuration to replace a route target (RT) with a VPN distinguisher, and shows the ingress ASBR configuration to replace the VPN distinguisher with a route target.

On the egress ASBR, IP extended community list 1 is configured to filter VPN routes by permitting only routes with RT 101:100. A route map named vpn-id-map1 says that any route that matches on routes that are allowed by IP extended community list 1 are subject to two **set** commands. The first **set** command deletes the RT from the route. The second **set** command sets the VPN distinguisher attribute to 111:100.

The **route-map vpn-id-map1 permit 20** command allows other routes, which are not part of the RT-to-VPN distinguisher mapping, to pass the route map so that they are not discarded. Without this command, the implicit deny would cause these routes to be discarded.

Finally, in autonomous system 2000, for the VPNv4 address family, the route map vpn-id-map1 is applied to routes going out to the neighbor at 192.168.101.1.

Egress ASBR

```
ip extcommunity-list 1 permit rt 101:100
!
route-map vpn-id-map1 permit 10
 match extcommunity 1
 set extcomm-list 1 delete
 set extcommunity vpn-distinguisher 111:100
!
route-map vpn-id-map1 permit 20
!
router bgp 2000
 neighbor 192.168.101.1 remote-as 2000
 address-family vpnv4
  neighbor 192.168.101.1 activate
  neighbor 192.168.101.1 route-map vpn-id-map1 out
  exit-address-family
!
```

On the ingress ASBR, IP extended community list 51 allows routes with a VPN distinguisher of 111:100. A route map named vpn-id-rewrite-map1 says that any route that matches on routes that are allowed by IP extended community list 51 are subject to two **set** commands. The first **set** command deletes the VPN distinguisher from the route. The second **set** command sets the RT to 101:1, and that RT is added to the RT list without replacing any RTs.

The **route-map vpn-id-rewrite-map1 permit 20** command allows other routes, which are not part of the VPN distinguisher-to-RT mapping, to pass the route map so that they are not discarded. Without this command, the implicit deny would cause those routes to be discarded.

Finally, in autonomous system 3000, for the VPNv4 address family, the route map named vpn-id-rewrite-map1 is applied to incoming routes destined for the neighbor at 192.168.0.81.

Ingress ASBR

```
ip extcommunity-list 51 permit vpn-distinguisher 111:100
!
route-map vpn-id-rewrite-map1 permit 10
 match extcommunity 51
 set extcomm-list 51 delete
 set extcommunity rt 101:1 additive
!
route-map vpn-id-rewrite-map1 permit 20
!
router bgp 3000
 neighbor 192.168.0.81 remote-as 3000
 address-family vpnv4
  neighbor 192.168.0.81 activate
  neighbor 192.168.0.81 route-map vpn-id-rewrite-map1 in
  exit-address-family
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP-VPN Distinguisher Attribute

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 135: Feature Information for BGP—VPN Distinguisher Attribute

Feature Name	Releases	Feature Information
BGP—VPN Distinguisher Attribute		The BGP—VPN Distinguisher Attribute feature allows a network administrator to keep source RTs private from an ASBR in a destination autonomous system. An RT at an egress ASBR is mapped to a VPN distinguisher, the VPN distinguisher is carried through the eBGP, and then it is mapped to an RT at the ingress ASBR. The following command was introduced: • **set extcommunity vpn-distinguisher** The following command was modified: • **show ip bgp vpnv4**

CHAPTER 106

BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard

The BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard feature introduces the ability to set a range of route target (RT) community attributes or VPN distinguisher community attributes when mapping them. A network administrator might want to map one or more RTs at an egress ASBR to different RTs at an ingress ASBR. The VPN Distinguisher Attribute feature allows an administrator to map RTs to a VPN distinguisher that is carried through an eBGP and then mapped to RTs at an ingress ASBR. The mapping is achieved by configuring a route map that sets an RT range or VPN distinguisher range of extended community attributes. Specifying a range rather than individual RTs saves time and simplifies the configuration. Furthermore, a VPN distinguisher range allows more than one VPN distinguisher attribute per route-map clause, thereby removing the restriction that applied prior to this feature.

Restrictions for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard

- A range (specified in the **set extcommunity rt** command or the **set extcommunity vpn-distinguisher** command) can include a maximum of 450 extended communities.
- The VPN distinguisher range is not relayed to an iBGP peer.

Information About BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

Benefits of RT and VPN Distinguisher Attribute Mapping Range

A network administrator might want to rewrite (or map) one or more route targets (RTs) at an egress ASBR to different RTs at an ingress ASBR. One use case would be to keep the RTs at the egress ASBR private from the ingress ASBR.

The rewrite is achieved by using inbound route maps, matching prefixes to route-map clauses that match inbound RTs, and mapping those RTs to different RTs recognized by the neighbor AS. Such a rewrite configuration could be complex on inbound route maps, with potentially hundreds of RTs that would need to be specified individually (configuring **set extcommunity rt** *value1 value2 value3 ...*). If the RTs being attached to the prefixes are consecutive, the configuration can be simplified by specifying a range of RTs. Thus, the benefits of the RT mapping range are saving time and simplifying the configuration.

Likewise, the mapping of RTs to a VPN distinguisher attribute (and vice versa) can also be simplified by specifying a range of RTs or VPN distinguishers. The BGP—VPN Distinguisher Attribute feature allows a network administrator to keep source RTs private from an ASBR in a destination AS. An RT at an egress ASBR is mapped to a VPN distinguisher, the VPN distinguisher is carried through the eBGP, and then it is mapped to an RT at the ingress ASBR.

The RT and VPN Distinguisher Attribute Mapping Range feature introduces the ability to specify a range of either route targets (RTs) or VPN distinguishers when mapping them.

Another benefit applies to setting a VPN distinguisher. Prior to this feature, only one **set extcommunity vpn-distinguisher** value was allowed per route-map clause. With the introduction of the mapping range, a range of VPN distinguishers can be set on a route.

How to Map RTs to RTs Using a Range

Replacing an RT with a Range of RTs

Perform this task on an egress ASBR to replace a route target (RT) with an RT range. Remember to replace the range of RTs with an RT on the ingress ASBR; that task is described in the “Replacing a Range of RTs with an RT” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *value*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**

8. **set extcommunity rt range** *start-value end-value*
9. **exit**
10. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **route-map** *map-tag* **out**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *value* **Example:** `Device(config)# ip extcommunity-list 22 permit rt 101:100`	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified RT are in the extended community list. • This example permits routes having RT 101:100 into the extended community list 22.
Step 4	**exit** **Example:** `Device(config-extcomm-list)# exit`	Exits the configuration mode and enters the next higher configuration mode.
Step 5	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map rt-mapping permit 10`	Configures a route map that permits or denies the routes allowed by the subsequent **match** command. • This example permits the routes allowed by the subsequent **match** command.
Step 6	**match extcommunity** *extended-community-list-name* **Example:** `Device(config-route-map)# match extcommunity 22`	Matches on the specified community list. • For this example, routes that match the extended community list 22 (which was configured in Step 3) are subject to the subsequent **set** commands.

	Command or Action	Purpose
Step 7	**set extcomm-list** *extcommunity-name* **delete** **Example:** `Device(config-route-map)# set extcomm-list 22 delete`	Deletes the RT from routes that are in the specified extended community list. • For this example, RTs are deleted from routes that are in extended community list 22.
Step 8	**set extcommunity rt range** *start-value end-value* **Example:** `Device(config-route-map)# set extcommunity rt range 500:1 500:9`	For the routes that are permitted by the route map, sets the specified RT range of extended community attributes, inclusive. • For this example, routes that match extended community 22 have their RT extended community attribute values set to 500:1, 500:2, 500:3, 500:4, 500:5, 500:6, 500:7, 500:8, and 500:9.
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map rt-mapping permit 20`	(Optional) Configures a route map entry that permits routes. • This example configures a route map entry that permits other routes not subject to the RT-to-RT range mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 12	**router bgp** *as-number* **Example:** `Device(config)# router bgp 3000`	Enters router configuration mode and creates a BGP routing process.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.103.1 remote-as 3000`	Specifies that the neighbor belongs to the autonomous system.
Step 14	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.

	Command or Action	Purpose
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.103.1 activate`	Activates the specified neighbor.
Step 16	**neighbor** *ip-address* **route-map** *map-tag* **out** **Example:** `Device(config-router-af)# neighbor 192.168.103.1 route-map rt-mapping out`	Applies the specified outgoing route map to the specified neighbor.
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters privileged EXEC mode.

Replacing a Range of RTs with an RT

Perform this task on an ingress ASBR to replace an RT range of attributes with an RT attribute. This task assumes you already configured the egress ASBR to replace the RT with an RT range; that task is described in the "Replacing an RT with a Range of RTs" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *reg-exp*
4. **exit**
5. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
6. **match extcommunity** *extended-community-list-name*
7. **set extcomm-list** *extcommunity-name* **delete**
8. **set extcommunity rt** *value* **additive**
9. **exit**
10. **route-map** *map-tag* {**permit** | **deny**} [*sequence-number*]
11. **exit**
12. **router bgp** *as-number*
13. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
14. **address-family vpnv4**
15. **neighbor** *ip-address* **activate**
16. **neighbor** *ip-address* **route-map** *map-tag* **in**
17. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip extcommunity-list** *expanded-list* {**permit** | **deny**} **rt** *reg-exp* **Example:** `Device(config)# ip extcommunity-list 128 permit rt 500:[1-9]`	Configures an IP extended community list to configure Virtual Private Network (VPN) route filtering, such that routes with the specified RT range are in the extended community list. • This example permits routes having RTs in the range 500:1 to 500:9 into the extended community list 128.
Step 4	**exit** **Example:** `Device(config-extcomm-list)# exit`	Exits the configuration mode and enters the next higher configuration mode.
Step 5	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map rtmap2 permit 10`	Configures a route map that permits or denies the routes allowed by the subsequent **match** command. • This example permits the routes allowed by the subsequent **match** command.
Step 6	**match extcommunity** *extended-community-list-name* **Example:** `Device(config-route-map)# match extcommunity 128`	Matches on the specified community list. • In this example, routes that match the extended community list 128 (which was configured in Step 3) are subject to the subsequent **set** commands.
Step 7	**set extcomm-list** *extcommunity-name* **delete** **Example:** `Device(config-route-map)# set extcomm-list 128 delete`	Deletes the RTs in the range from routes that are in the specified extended community list. • In this example, RTs in the range are deleted from routes that are in extended community list 128.
Step 8	**set extcommunity rt** *value* **additive** **Example:** `Device(config-route-map)# set extcommunity rt 400:1 additive`	Sets the routes that are permitted by the route map with the specified RT. • In this example, routes that match extended community 128 have their RT set to 400:1. The **additive** keyword causes the RT to be added to the RT list without replacing any RTs.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 10	**route-map** *map-tag* {**permit** | **deny**} [*sequence-number*] **Example:** `Device(config)# route-map rtmap2 permit 20`	(Optional) Configures a route map entry that permits routes. • This example configures a route map entry that permits other routes not subject to the RT-range-to-RT mapping. If you do not perform this step, all other routes are subject to an implicit deny.
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 12	**router bgp** *as-number* **Example:** `Device(config)# router bgp 4000`	Enters router configuration mode and creates a BGP routing process.
Step 13	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 192.168.0.50 remote-as 4000`	Specifies that the neighbor belongs to the autonomous system.
Step 14	**address-family vpnv4** **Example:** `Device(config-router-af)# address-family vpnv4`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 15	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.0.50 activate`	Activates the specified neighbor.
Step 16	**neighbor** *ip-address* **route-map** *map-tag* **in** **Example:** `Device(config-router-af)# neighbor 192.168.0.50 route-map rtmap2 in`	Applies the specified incoming route map to the specified neighbor.

	Command or Action	Purpose
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and enters privileged EXEC mode.

Configuration Examples for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

Example: Replacing an RT with a Range of RTs

In the following example, on the egress ASBR, routes having RT 101:100 are in the extended community list 22. A route-map named rt-mapping matches on extended community list 22 and deletes the RT from routes in the community list. Routes that match the community list have their RT set to an RT in the range from 500:1 to 500:9. The route map is applied to the neighbor 192.168.103.1.

Egress ASBR

```
ip extcommunity-list 22 permit rt 101:100
!
route-map rt-mapping permit 10
 match extcommunity 22
 set extcomm-list 22 delete
 set extcommunity rt range 500:1 500:9
!
route-map rt-mapping permit 20
!
router bgp 3000
 neighbor 192.168.103.1 remote-as 3000
 address-family vpnv4
  neighbor 192.168.103.1 activate
  neighbor 192.168.103.1 route-map rt-mapping out
  exit-address-family
!
```

On the ingress ASBR, RTs in the range 500:1 to 500:9 belong to extended community list 128. A route map named rtmap2 maps those RTs to RT 400:1. The route map is applied to the neighbor 192.168.0.50.

Ingress ASBR

```
ip extcommunity-list 128 permit RT:500:[1-9]
!
route-map rtmap2 permit 10
 match extcommunity 128
 set extcomm-list 128 delete
 set extcommunity rt 400:1 additive
!
route-map rtmap2 permit 20
```

```
!
router bgp 4000
 neighbor 192.168.0.50 remote-as 4000
 address-family vpnv4
  neighbor 192.168.0.50 activate
  neighbor 192.168.0.50 route-map rtmap2 in
  exit-address-family
!
```

Example: Replacing an RT with a Range of VPN Distinguishers

In the following example, on the egress ASBR, routes having RT 201:100 are in the extended community list 22. A route-map named rt-mapping matches on extended community list 22 and deletes the RT from routes in the community list. Routes that match the community list have their VPN distinguishers set to VPN distinguishers in the range from 600:1 to 600:8. The route map is applied to the neighbor 192.168.103.1.

Egress ASBR

```
ip extcommunity-list 22 permit rt 201:100
!
route-map rt-mapping permit 10
 match extcommunity 22
 set extcomm-list 22 delete
 set extcommunity vpn-distinguisher range 600:1 600:8
!
route-map rt-mapping permit 20
!
router bgp 3000
 neighbor 192.168.103.1 remote-as 3000
 address-family vpnv4
  neighbor 192.168.103.1 activate
  neighbor 192.168.103.1 route-map rt-mapping out
  exit-address-family
!
```

On the ingress ASBR, VPN distinguishers in the range 600:1 to 600:8 belong to extended community list 101. A route map named rtmap2 maps those VPN distinguishers to RT range 700:1 700:10. The route map is applied to the neighbor 192.168.0.50. The additive option adds the new range to the existing value without replacing it.

Ingress ASBR

```
ip extcommunity-list 101 permit VD:600:[1-8]
!
route-map rtmap2 permit 10
 match extcommunity 101
 set extcomm-list 101 delete
 set extcommunity rt 700:1 700:10 additive
!
route-map rtmap2 permit 20
!
router bgp 4000
 neighbor 192.168.0.50 remote-as 4000
 address-family vpnv4
  neighbor 192.168.0.50 activate
  neighbor 192.168.0.50 route-map rtmap2 in
```

```
  exit-address-family
!
```

Additional References for BGP-RT and VPN Distinguisher Attribute Rewrite Wildcard

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP—VPN Distinguisher Attribute	"BGP—VPN Distinguisher Attribute" module in the *IP Routing: BGP Configuration Guide, Cisco IOS XE Release 3S*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 136: Feature Information for BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard

Feature Name	Releases	Feature Information
BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard		The BGP—RT and VPN Distinguisher Attribute Rewrite Wildcard feature introduces the ability to set a range of route target (RT) community attributes or VPN distinguisher community attributes when mapping them. A network administrator might want to map one or more RTs at an egress ASBR to different RTs at an ingress ASBR. The VPN Distinguisher Attribute feature allows an administrator to map RTs to a VPN distinguisher that is carried through an eBGP and then mapped to RTs at an ingress ASBR. The mapping is achieved by configuring a route map that sets an RT range or VPN distinguisher range of extended community attributes. Specifying a range rather than individual RTs saves time and simplifies the configuration. Furthermore, a VPN distinguisher range allows more than one VPN distinguisher attribute per route-map clause, thereby removing the restriction that applied prior to this feature. The following commands were modified: • **set extcommunity rt** • **set extcommunity vpn-distinguisher**

CHAPTER 107

VPLS BGP Signaling

The two primary functions of the Virtual Private LAN Service (VPLS) control plane are autodiscovery and signaling. The VPLS BGP Signaling feature enables you to use BGP as both an autodiscovery and a signaling protocol for VPLS, in accordance with RFC 4761.

Prerequisites for VPLS BGP Signaling

You are familiar with the concepts in the “Configuring Virtual Private LAN Services” and the “VPLS Autodiscovery BGP Based” modules of the .

Information About VPLS BGP Signaling

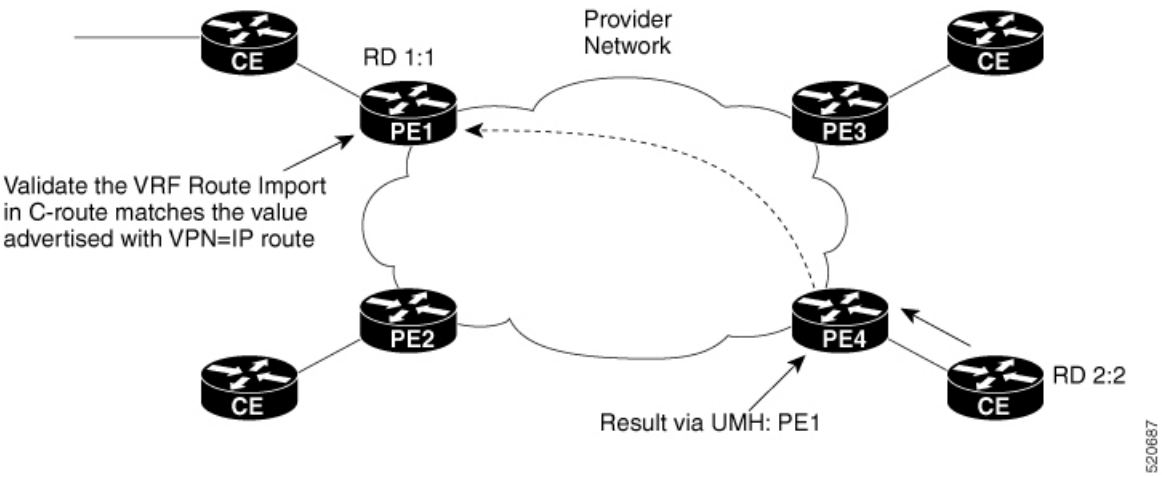
Overview of VPLS BGP Signaling

Prior to the VPLS BGP Signaling feature, BGP was used for autodiscovery and Label Distribution Protocol (LDP) for signaling in accordance with RFC 6074. The VPLS BGP Signaling feature enables you to use BGP as the control plane protocol for both autodiscovery and signaling in accordance with RFC 4761.

As specified in RFC 4761, internal BGP (iBGP) peers will exchange update messages of the L2VPN AFI/SAFI with L2VPN information to perform both autodiscovery and signaling. The BGP multiprotocol Network Layer Reachability Information (NLRI) consists of a Route Distinguisher (RD), VPLS Endpoint ID (VE ID), VE Block Offset (VBO), VE Block Size (VBS), and Label Base (LB).

The figure below shows the format of the NLRI for RFC 4761.

Figure 122: RFC 4761 NLRI



Additional information, such as next-hop, route target (specified for a VPLS instance), and other Layer 2 data are carried in the BGP extended community attributes. A route target-based import/export mechanism similar to L3VPN is performed by BGP to filter L2VPN NLRIs of a particular VPLS instance.

Whether you use BGP signaling (RFC 4761) or LDP signaling (RFC 6074) depends on the commands you specify. To enable the VPLS BGP Signaling feature, use the **autodiscovery bgp signaling bgp** command in L2 VFI configuration mode. This command is supported on a per VPLS instance basis.

If a BGP session receives an invalid (that is, not matching the configuration) BGP update advertisement (update or withdraw), it is ignored.

BGP's main task in supporting VPLS is route distribution via the L2VPN address family and interactions with L2VPN. Interactions between BGP and other components remain the same. Basic BGP functionalities like best-path selection, next-hop handling, and update generation, continue to operate in the same manner with VPLS BGP signaling. BGP RT constraint works seamlessly with the BGP VPLS Signaling feature.

The above example shows sample configuration on one PE. Similar configuration can be mirrored on other PEs.

How to Configure VPLS BGP Signaling

Configuring VPLS BGP Signaling

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2vpn vfi context** *name*
4. **vpn id** *vpn-id*
5. **autodiscovery bgp signaling** {**bgp** | **ldp**} [**template** *template-name*]
6. **ve id** *ve-id*
7. **ve range** *ve-range*

8. **exit**
9. **exit**
10. **router bgp** *autonomous-system-number*
11. **bgp graceful-restart**
12. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
13. **address-family l2vpn** [**vpls**]
14. **neighbor** *ip-address* **activate**
15. **neighbor** *ip-address* **send-community** [**both** | **standard** | **extended**]
16. **neighbor** *ip-address* **suppress-signaling-protocol ldp**
17. **end**
18. **show bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**l2vpn vfi context** *name* **Example:** `Device(config)# l2vpn vfi context vfi1`	Establishes a L2VPN virtual forwarding interface (VFI) between two or more separate networks and enters Layer 2 VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Device(config-vfi)# vpn id 100`	Configures a VPN ID for the VPLS domain.
Step 5	**autodiscovery bgp signaling** {**bgp**	**ldp**} [**template** *template-name*] **Example:** `Device(config-vfi)# autodiscovery bgp signaling bgp`	Enables BGP signaling and discovery or LDP signaling and enters L2VPN VFI autodiscovery configuration mode. **Note** For the VPLS BGP Signaling feature use the **autodiscovery bgp signaling bgp** command.
Step 6	**ve id** *ve-id* **Example:** `Device(config-vfi-autodiscovery)# ve id 1001`	Specifies the VPLS endpoint (VE) device ID value. The VE ID identifies a VFI within a VPLS service. The VE device ID value is from 1 to 16384.

	Command or Action	Purpose
Step 7	**ve range** *ve-range* **Example:** `Device(config-vfi-autodiscovery)# ve range 12`	Specifies the VE device ID range value. The VE range overrides the minimum size of VE blocks. The default minimum size is 10. Any configured VE range must be higher than 10.
Step 8	**exit** **Example:** `Device(config-vfi-autodiscovery)# exit`	Exits L2VPN VFI autodiscovery configuration mode and enters L2VPN VFI configuration mode.
Step 9	**exit** **Example:** `Device(config-vfi)# exit`	Exits L2VPN VFI configuration mode and enters global configuration mode.
Step 10	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode to create or configure a BGP routing process.
Step 11	**bgp graceful-restart** **Example:** `Device(config-router)# bgp graceful-restart`	Enables the BGP graceful restart capability and BGP nonstop forwarding (NSF) awareness.
Step 12	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 10.10.10.1 remote-as 100`	Configures peering with a BGP neighbor in the specified autonomous system.
Step 13	**address-family l2vpn** [**vpls**] **Example:** `Device(config-router)# address-family l2vpn vpls`	Specifies the L2VPN address family and enters address family configuration mode. • The optional **vpls** keyword specifies that VPLS endpoint provisioning information is to be distributed to BGP peers. In this example, an L2VPN VPLS address family session is created.
Step 14	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.10.10.1 activate`	Enables the neighbor to exchange information for the L2VPN VPLS address family with the local device.

	Command or Action	Purpose
Step 15	**neighbor** *ip-address* **send-community** [**both** | **standard** | **extended**] **Example:** `Device(config-router-af)# neighbor 10.10.10.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • In this example, an extended communities attribute is sent to the neighbor at 10.10.10.1.
Step 16	**neighbor** *ip-address* **suppress-signaling-protocol ldp** **Example:** `Device(config-router-af)# neighbor 10.10.10.1 suppress-signaling-protocol ldp`	Suppresses LDP signaling and enables BGP signaling. • In this example LDP signaling is suppressed (and BGP signaling enabled) for the neighbor at 10.10.10.1.
Step 17	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 18	**show bgp l2vpn vpls** {**all** | **rd** *route-distinguisher*} **Example:** `Device# show bgp l2vpn vpls all`	(Optional) Displays information about the L2VPN VPLS address family.

Configuration Examples for VPLS BGP Signaling

Example: Configuring and Verifying VPLS BGP Signaling

```
l2vpn vfi context vfi1
 vpn id 100
 autodiscovery bgp signaling bgp
  ve id 1001
  ve range 10
 !
!
router bgp 100
 bgp graceful-restart
 neighbor 209.165.200.224 remote-as 100
 neighbor 209.165.200.224 update-source Loopback1
 !
 address-family l2vpn vpls
  neighbor 209.165.200.224 activate
  neighbor 209.165.200.224 send-community extended
  neighbor 209.165.200.224 suppress-signaling-protocol ldp
  exit-address-family
 !
show bgp l2vpn vpls all

Network                          Next Hop            Metric LocPrf Weight Path
```

```
Route Distinguisher: 100:100
*>100:100:VEID-1001:Blk-1001/136    0.0.0.0                              32768  ?

*>i 100:100:VEID-1003:Blk-1000/136  209.165.200.224              0      100     0     ?
```

Additional References for VPLS BGP Signaling

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples.	Cisco IOS IP Routing: BGP Command Reference
Configuring Virtual Private LAN Services	
Configuring Access Port	Configuring Virtual Private LAN Services,
VPLS Autodiscovery BGP Based	

Standards and RFCs

Standard/RFC	Title
RFC 4761	*Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*
RFC 6074	*Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VPLS BGP Signaling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 137: Feature Information for VPLS BGP Signaling

Feature Name	Releases	Feature Information
VPLS BGP Signaling		The VPLS BGP Signaling feature enables you to use BGP as both an autodiscovery and signaling protocol for VPLS, in accordance with RFC 4761. The following commands were introduced or modified: **autodiscovery (MPLS)**, **neighbor suppress-signaling-protocol**, **show bgp l2vpn vpls**, and **ve**.

CHAPTER 108

Multicast VPN BGP Dampening

A single receiver in a specific multicast group or a group of receivers that are going up and down frequently and interested in a specific multicast group activates the Multicast VPN BGP Dampening feature to dampen type 7 routes (C-multicast route Join/Prune) within the core using BGP signaling. The feature reduces the churn caused by customer-side join/prune requests to avoid unnecessary BGP MVPN type 6/7 C-route control information.

Prerequisites for Multicast VPN BGP Dampening

- You understand the concepts in the “BGP Route Dampening” module of the *IP Routing: BGP Configuration Guide*.

Information About Multicast VPN BGP Dampening

Overview of Multicast VPN BGP Dampening

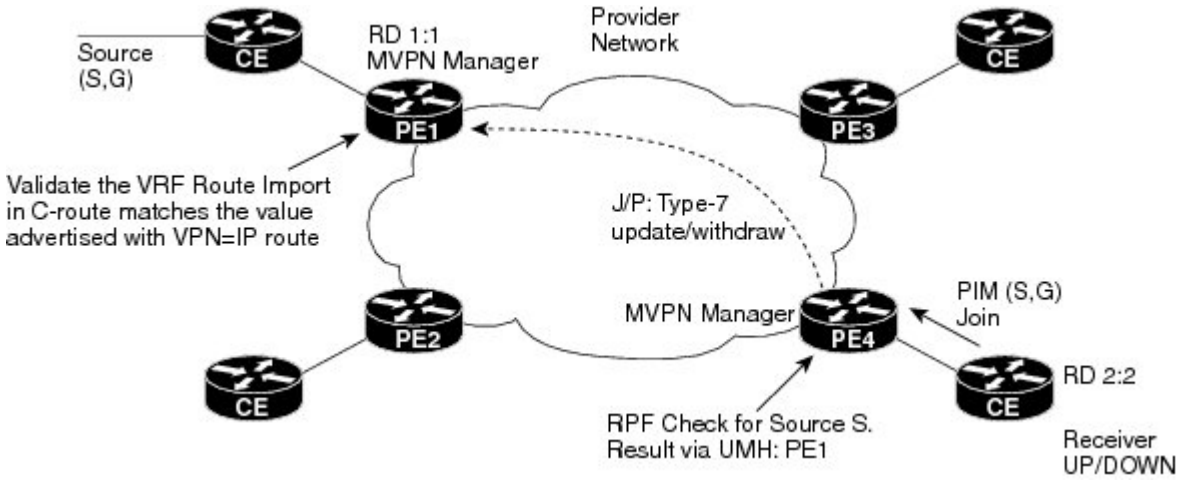
BGP Route Dampening

Route dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when its availability alternates repeatedly. Cisco devices that are running BGP contain a mechanism designed to “dampen” the destabilizing effect of flapping routes. When a Cisco device running BGP detects a flapping route, it automatically dampens that route.

The figure below shows illustrates the Multicast VPN BGP dampening mechanism.

Multicast VPN BGP Dampening

Figure 123: Multicast VPN BGP Dampening



A single receiver in a multicast group or a group of receivers that are flapping frequently and interested in a specific multicast group actrivates multicast VPN (MVPN) BGP dampening. MVPN BGP dampening dampens the type 7 multicast routes (customer-multicast, or "C-multicast," route join/prune) within the core using BGP signaling.

When MVPN BGP dampening is not enabled, the source sends data even though the receiver may be down. When the receiver is down, there is no periodic 60-second C-PIM join towards the provider edge (PE) device causing the PIM to timeout on the PE side after the default period (three minutes). The MVPN manager sends a prune message to BGP, which is a type 7 route (C-multicast route withdraw).

When the receiver is up, it sends a new (S,G) join request to the customer edge (CE) device. The C-PIM join is received by the PE device and a new type 7 C-multicast update is sent by BGP to the auto-discovered MVPN peers. The upstream multicast peer converts the BGP type 7 update to a PIM join to the source, and the source sends the data traffic that the receiver should receive via the downstream PE using the MDT tunnel. If the receiver goes up and down frequently, the source side PIM receives join/prune messages frequently and can cause the source to respond accordingly.

When MVPN BGP dampening is enabled, the general dampening mechanism in BGP will be applied to MVPN VRF instances. Join/Prune messages from the CE side are sent from an MVPN manager as updates/withdraw to the MVPN PE device. The MVPN manager on PE devices send join/prune messages to the customer side for Reverse Path Forwarding (RPF) and upstream multihop (UMH) nexthop changes.

How to Configure Multicast VPN BGP Dampening

Configuring Multicast VPN BGP Dampening

Perform this task to enable and configure multicast VPN BGP dampening.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family** [**ipv4** | **ipv6**] **mvpn vrf** *vrf-name*
5. **bgp dampening** [*half-life reuse suppress max-suppress-time*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family** [**ipv4**	**ipv6**] **mvpn vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 mvpn vrf blue`	Specifies the address family and enters address family configuration mode. • Use the **ipv4** keyword to enable IPv4 multicast C-route exchange. • Use the **ipv6** keyword to enable IPv6 multicast C-route exchange. **Note** The **vrf** keyword and *vrf-name* argument must be specified at this point to enable multicast VPN BGP dampening in the next step.
Step 5	**bgp dampening** [*half-life reuse suppress max-suppress-time*] **Example:** `Device(config-router-af)# bgp dampening 30 1500 10000 120`	Enables BGP route dampening and changes the default values of route dampening factors. The *half-life*, *reuse*, *suppress*, and *max-suppress-time* arguments are all position dependent; if one argument is entered, then all the arguments must be entered. **Note** Repeat steps 4 and 5 to enable multicast VPN BGP dampening on alternative VRFs.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Monitoring and Maintaining Multicast VPN BGP Dampening

Perform the steps in this task as required to monitor and maintain multicast VPN BGP dampening.

SUMMARY STEPS

1. **enable**
2. **show bgp** {**ipv4** | **ipv6**} **mvpn** {**all** | **rd** *route-distinguisher* | **vpn** *vrf-name*} [**dampening** {**dampened-paths** | **flap-statistics** [**filter-list** *access-list-number* | **quote-regexp** *regexp* | **regexp** *regexp*]}]
3. **clear ip bgp** {**ipv4** | **ipv6**} **mvpn vrf** *vrf-name* {**dampening** | **flap-statistics**}

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show bgp** {**ipv4** | **ipv6**} **mvpn** {**all** | **rd** *route-distinguisher* | **vpn** *vrf-name*} [**dampening** {**dampened-paths** | **flap-statistics** [**filter-list** *access-list-number* | **quote-regexp** *regexp* | **regexp** *regexp*]}]

Use this command to monitor multicast VPN BGP dampening.

- The **dampened-path** keyword displays information about BGP dampened routes.
- The **parameters** keyword displays detailed BGP dampening information.
- The **flap-statistics** keyword displays information on BGP flap statistics.

Example:

```
Device# show bgp ipv4 mvpn vrf blue route-type 7 111.111.111.111:11111 55 202.100.0.6 232.1.1.1

BGP routing table entry for [7][111.111.111.111:11111][55][202.100.0.6/32][232.1.1.1/32]/22, version
 17
Paths: (1 available, no best path)
Flag: 0x820
  Not advertised to any peer
  Refresh Epoch 1
  Local, (suppressed due to dampening)
    0.0.0.0 from 0.0.0.0 (205.3.0.3)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local
      Extended Community: RT:205.1.0.1:1
      Dampinfo: penalty 3472, flapped 4 times in 00:04:42, reuse in 00:00:23
      rx pathid: 0, tx pathid: 0
```

Step 3 **clear ip bgp** {**ipv4** | **ipv6**} **mvpn vrf** *vrf-name* {**dampening** | **flap-statistics**}

Use this command to clear the accumulated penalty for routes that are received on a router that has multicast VPN BGP dampening enabled.

- The **dampening** keyword clears multicast VPN BGP dampening information.
- The **flap-statistic** keyword clears multicast VPN BGP dampening flap statistics.

Example:

```
Device# clear ip bgp ipv4 mvpn vrf blue dampening
```

Configuration Examples for Multicast VPN BGP Dampening

Example: Configuring Multicast VPN BGP Dampening

The following example shows multicast VPN BGP dampening is applied to the VRFs named blue and red, but not to the VRF named green:

```
address-family ipv4 mvpn vrf blue
 bgp dampening

address-family ipv4 mvpn vrf red
 bgp dampening

address-family ipv4 mvpn vrf green
 no bgp dampening
```

Additional References for Multicast VPN BGP Dampening

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP route dampening	"BGP Route Dampening" section of the "Configuring Internal BGP Features" module in the *IP Routing: BGP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2439	*BGP Route Flap Damping*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multicast VPN BGP Dampening

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 138: Feature Information for Multicast VPN BGP Dampening

Feature Name	Releases	Feature Information
Multicast VPN BGP Dampening	Cisco IOS XE Release 3.8S	A single receiver in a specific multicast group or a group of receivers that are going up and down frequently and interested in a specific multicast group will cause the Multicast VPN BGP Dampening feature to dampen type 7 routes (C-multicast route join/prune) within the core using BGP signaling. The following commands were introduced or modified: **address-family mvpn**, **clear ip bgp mvpn**, **show bgp mvpn**, and **show ip bgp ipv4**.

CHAPTER 109

BGP—IPv6 NSR

Border Gateway Protocol (BGP) support for Nonstop Routing (NSR) enables provider edge (PE) routers to maintain BGP state with customer edge (CE) routers and ensure continuous packet forwarding during a Route Processor (RP) switchover or during a planned In-Service Software Upgrade (ISSU) for a PE router. The BGP—IPv6 NSR feature extends BGP support for NSR to Cisco IPv6 VPN Provider Edge Routers (6VPE).

Prerequisites for BGP—IPv6 NSR

- Your network is configured to run BGP.
- Multiprotocol Layer Switching (MPLS) Layer 3 Virtual Private Networks (VPNs) are configured.
- All platforms are HA capable.
- You are familiar with the concepts in the "BGP Support for Nonstop Routing (NSR) with Stateful Switchover (SSO)" and "BGP NSR Support for iBGP Peers" modules of the *IP Routing: BGP Configuration Guide*.

Information About BGP—IPv6 NSR

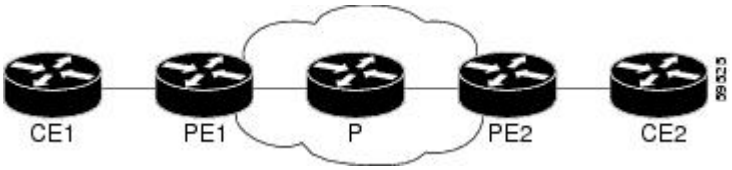
Overview of BGP—IPv6 NSR

Nonstop routing (NSR) is beneficial for BGP peers because it reduces the likelihood of dropped packets during switchover from the active Route Processor (RP) to the standby RP. Switchover occurs when the active RP fails for some reason and the standby RP takes control of active RP operations. The BGP—IPv6 NSR feature extends BGP support for NSR to include the following IPv6-based address families:

- IPv6 unicast

- IPv6 unicast + label
- IPv6 PE-CE
- VPNv6 unicast

Figure 124: Basic 6VPE Network Configuration



The figure above depicts a basic deployment scenario. Provider edge (PE) router 1, P, and PE2 form a 6VPE cloud. The customer edge (CE) router 1 to PE1 connection is IPv6 (VRF). The PEs are HA/SSO and NSF capable. The P routers are capable of Multiprotocol Label Switching (MPLS) label preservation (NSF equivalent).

As the CE1 is customer equipment, the provider cannot determine that it must be upgraded to be NSF aware. If PE1 can perform NSR on its connection to CE1, then CE1 will not be aware or impacted when PE1 performs a switchover in SSO mode. For all other connections within the autonomous system, the operations may be NSF or graceful restart. This means the control plane will be reset, and all the immediate peers will be aware of it and will resend data to help re-establish the session, but forwarding will be uninterrupted.

Neighbors not operating under NSR are still expected to be NSF capable/aware. If the CE is already NSF aware (that is, it can handle a BGP graceful restart by its peers), then the PE-CE connection will not be NSR, and will instead follow the regular NSF processing model. This parallels NSR for VPNv4 and assists in conserving network resources.

How to Configure BGP—IPv6 NSR

Configuring BGP—IPv6 NSR

Perform this task on a PE router if you want to configure a BGP peer to support BGP—IPv6 NSR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Enter one of the following:
   - **address-family ipv6** [**unicast** | **multicast** | **vpnv6**] [**vrf** *vrf-name*]
   - **address-family vpnv6** [**unicast** | **multicast**]
5. **neighbor** *ipv6-address%* **remote-as** *as-number*
6. **neighbor** *ipv6-address%* **activate**
7. **neighbor** *ipv6-address%* **ha-mode sso**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 4000`	Enters router configuration mode for the specified routing process.
Step 4	Enter one of the following: • **address-family ipv6** [**unicast** | **multicast** | **vpnv6**] [**vrf** *vrf-name*] • **address-family vpnv6** [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6 unicast`	Specifies the IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv6 unicast address family. • The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv6 address family configuration mode commands.
Step 5	**neighbor** *ipv6-address%* **remote-as** *as-number* **Example:** `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 remote-as 4000`	Specifies the autonomous system of the neighbor.
Step 6	**neighbor** *ipv6-address%* **activate** **Example:** `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 activate`	Activates the specified peer.
Step 7	**neighbor** *ipv6-address%* **ha-mode sso** **Example:** `Device(config-router-af)# neighbor 2001:DB8:0:CC00::1 ha-mode sso`	Configures a BGP neighbor to support BGP NSR.
Step 8	**end** **Example:**	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router-af)# end`	

Configuration Examples for BGP—IPv6 NSR

Example: Configuring BGP—IPv6 NSR

```
router bgp 4000
 address-family ipv6 unicast
 neighbor 2001:DB8:0:CC00::1 remote-as 4000
 neighbor 2001:DB8:0:CC00::1 activate
 neighbor 2001:DB8:0:CC00::1 ha-mode sso
```

Additional References for BGP—IPv6 NSR

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP—IPv6 NSR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 139: Feature Information for BGP—IPv6 NSR

Feature Name	Releases	Feature Information
BGP—IPv6 NSR	Cisco IOS XE Release 3.9S	BGP support for NSR enables provider edge (PE) routers to maintain BGP state with customer edge (CE) routers and ensure continuous packet forwarding during a Route Processor (RP) switchover or during a planned ISSU for a PE router. The BGP—IPv6 NSR feature extends BGP support for NSR to Cisco IPv6 VPN Provider Edge Routers (6VPE).

CHAPTER 110

BGP-VRF-Aware Conditional Advertisement

The Border Gateway Protocol (BGP) VRF-Aware Conditional Advertisement feature provides additional control of the advertisement of routes and extends this control to within a virtual routing and forwarding (VRF) instance.

Information About BGP VRF-Aware Conditional Advertisement

VRF-Aware Conditional Advertisement

The Border Gateway Protocol (BGP) VRF-Aware Conditional Advertisement feature provides additional control of the advertisement of routes and extends this control within a virtual routing and forwarding (VRF) instance.

BGP Conditional Advertisement

Normally, routes are propagated regardless of the existence of a different route. The BGP conditional advertisement feature uses the **exist-map**, **non-exist-map**, and the **advertise-map** keywords of the **neighbor** command in order to track routes by the route prefix. If a route prefix is not present in output of the **non-exist-map** command, then the route specified by the **advertise-map** is announced. This feature is useful for multihomed networks, in which some prefixes are advertised to one of the providers only if information from the other provider is not present (this indicates a failure in the peering session or partial reachability). The conditional BGP announcements are sent in addition to the normal announcements that a BGP router sends to its peers.

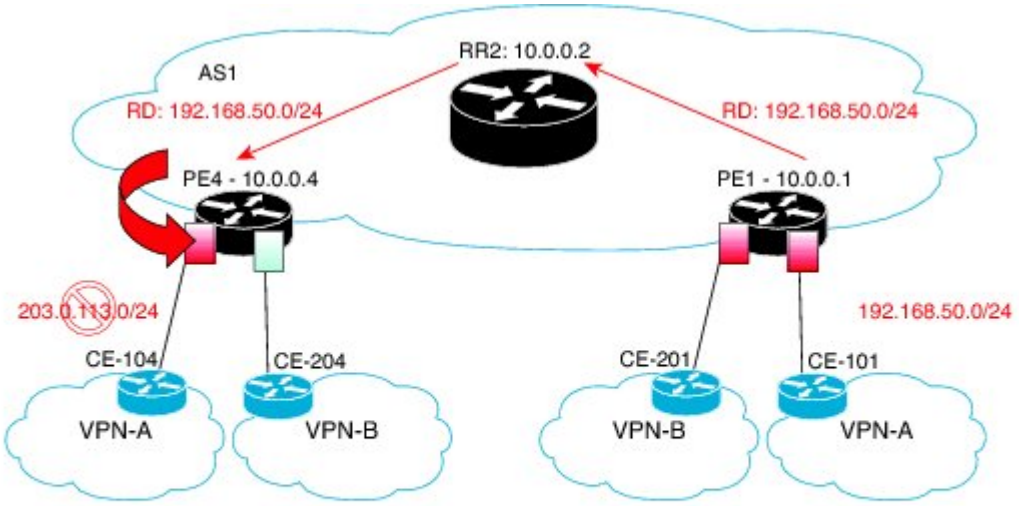
VRF-Aware Conditional Advertisement

This feature extends support for BGP VRF-aware conditional advertisement to the following address families:

- IPv4 unicast
- IPv4 unicast VRF

- IPv6 unicast
- IPv6 unicast VRF

Figure 125: VRF-Based Conditional Advertisement



PE4: VRF RED Routing Table	
EXIST	192.168.50.0/24
ADVERT	203.0.113.0/24

336577

The figure above shows the IPv4 prefix 192.168.50.0/24 being advertised by a remote CE101 into VRF RED on PE1. The prefix flows as a MP-BGP VPN prefix and is imported into the VRF RED on PE4. On the PE4 the conditions configured by the **exist-map** command relating to this prefix in the BGP VRF RED table becomes the condition to advertise the prefix 203.0.113.0/24 to the CE104, that is, peer-activated under the VRF RED on the PE4. This scenario assumes that 203.0.113.0/24 is in the VRF RED BGP table. If 203.0.113.0/24 is not in the table, this policy is ignored.

- If 192.168.50.0/24 exists in PE4's BGP table, then the 203.0.113.0/24 network is advertised to CE104.
- If 192.168.50.0/24 does not exist in PE4's BGP table, then the 203.0.113.0/24 network is not advertised to CE104.

How to Configure BGP VRF-Aware Conditional Advertisement

Configuring BGP VRF-Aware Conditional Advertisement

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Enter one of the following:

- **address-family ipv4** [**unicast**] [**vrf** *vrf-name*]
- **address-family ipv6** [**unicast**] [**vrf** *vrf-name*]

5. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *ipv6-address*} **activate**
7. **neighbor** {*ip-address* | *ipv6-address*} **advertise-map** *map-name* {**exist-map** *map-name* | **non-exist-map** *map-name*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	Enter one of the following: • **address-family ipv4** [**unicast**] [**vrf** *vrf-name*] • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 vrf VRFRED`	Specifies the IPv4 or IPv6 address family and enters address family configuration mode. • The **unicast** keyword specifies the IPv4 or IPv6 unicast address family. • The **vrf** keyword and *vrf-name* argument specify the name of the virtual routing and forwarding (VRF) instance to associate with subsequent IPv4 or IPv6 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *ipv6-address*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.0.2.1 remote-as 104`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 or IPv6 multiprotocol BGP neighbor table of the local device.
Step 6	**neighbor** {*ip-address* | *ipv6-address*} **activate** **Example:** `Device(config-router-af)# neighbor 192.0.2.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 or IPv6 address family with the local device.

	Command or Action	Purpose
Step 7	**neighbor** {*ip-address* | *ipv6-address*} **advertise-map** *map-name* {**exist-map** *map-name* | **non-exist-map** *map-name*} **Example:** `Device(config-router-af)# neighbor 192.0.2.1 advertise-map ADV-1 exist-map EXIST-1`	Enables conditional advertisement towards a neighbor to allow the advertisement of prefixes mapped by the **advertise-map** command based on the criteria defined under **exist** or **non-exist** maps. • The **advertise-map** *map-name* keyword-argument pair specifies the name of the route map used to define the advertised routes. • The **exist-map** *map-name* keyword-argument pair specifies the condition that can be satisfied by a set of routes in the BGP table. If the condition is satisfied then the routes in the BGP table matching those specified in **advertise map** will be advertised. If the routes matching those specified in **exist-map** do not exist in the BGP table, those routes will not be advertised. • The **non-exist-map** *map-name* keyword-argument pair specifies the condition that is compared to a set of routes in the BGP table. If the routes in the **non-exist-map** are not present in the BGP table, then the routes matching those specified in **advertise map** will be advertised. If the routes matching those specified in **non-exist-map** are present in the BGP table, then the routes matching **advertise-map** will not be advertised.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

What to do next

To verify the configuration of the BGP VRF-Aware Conditional Advertisement feature, use the **show bgp ip neighbors** command.

Configuration Examples for BGP VRF-Aware Conditional Advertisement

Example: Configuring BGP VRF-Aware Conditional Advertisement

The following examples use the configuration in figure 1:

CE 101: The source of the prefixes

```
router bgp 101
 bgp log-neighbor-changes
 timers bgp 0 0
 neighbor 172.16.1.2 remote-as 65000
 !
 address-family ipv4
  network 21.21.21.0 mask 255.255.255.0
  network 22.22.22.22 mask 255.255.255.255
  network 31.0.0.0
  network 33.0.0.0
  network 44.0.0.0
  network 192.0.254 mask 255.255.255.0
  network 192.0.2.50
  neighbor 172.16.1.3 activate
 exit-address-family
```

PE 1

```
router bgp 65000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 timers bgp 0 0
 neighbor 10.0.0.2 remote-as 65000
 neighbor 10.0.0.2 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf blue
  neighbor 198.51.100.10 remote-as 201
  neighbor 198.51.100.10 activate
 exit-address-family
 !
 address-family ipv4 vrf red
  neighbor 172.16.1.2 remote-as 101
  neighbor 172.16.1.2 activate
 exit-address-family
```

PE 4

```
router bgp 65000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 timers bgp 0 0
 neighbor 10.0.0.2 remote-as 65000
 neighbor 10.0.0.2 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.2 activate
  neighbor 10.0.0.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blue
```

```
  neighbor 198.51.100.12 remote-as 204
  neighbor 198.51.100.12 activate
 exit-address-family
 !
 address-family ipv4 vrf red
  neighbor 198.51.100.3 remote-as 104
  neighbor 198.51.100.3 activate
  neighbor 198.51.100.3 advertise-map ADV-1 exist-map EXIST-1
  neighbor 198.51.100.3 advertise-map ADV-2 exist-map EXIST-2
  neighbor 198.51.100.3 advertise-map ADV-3 exist-map EXIST-3
  neighbor 198.51.100.3 advertise-map ADV-4 exist-map EXIST-4
 exit-address-family
!
ip prefix-list pl-adv-1 seq 5 permit 22.22.22.22/32
!
ip prefix-list pl-adv-2 seq 5 permit 44.0.0.0/8 □
!
ip prefix-list pl-adv-3 seq 5 permit 33.0.0.0/8
!
ip prefix-list pl-adv-4 seq 5 permit 128.16.16.0/24
!
ip prefix-list pl-exist-1 seq 5 permit 21.21.21.0/24
!
ip prefix-list pl-exist-2 seq 5 permit 41.0.0.0/8 □
!
ip prefix-list pl-exist-3 seq 5 permit 31.0.0.0/8
!
ip prefix-list pl-exist-4 seq 5 permit 192.168.50.0/24
!
route-map EXIST-4 permit 10
 match ip address prefix-list pl-exist-4
!
route-map ADV-4 permit 10
 match ip address prefix-list pl-adv-4
!
route-map EXIST-2 permit 10
 match ip address prefix-list pl-exist-2
!
route-map ADV-2 permit 10
 match ip address prefix-list pl-adv-2
!
route-map EXIST-3 permit 10
 match ip address prefix-list pl-exist-3
!
route-map ADV-3 permit 10
 match ip address prefix-list pl-adv-3
!
route-map EXIST-1 permit 10
 match ip address prefix-list pl-exist-1
!
route-map ADV-1 permit 10
 match ip address prefix-list pl-adv-1
```

Example: Verifying BGP VRF-Aware Conditional Advertisement

The following examples use the configuration in figure 1:

CE 101

```
CE101# show ip bgp all
```

```
For address family: IPv4 Unicast
BGP table version is 28, local router ID is 203.0.113.11
     Network          Next Hop            Metric LocPrf Weight Path
 *>  21.21.21.0/24     0.0.0.0                  0         32768 i
 *>  22.22.22.22/32    0.0.0.0                  0         32768 i
 *>  31.0.0.0          0.0.0.0                  0         32768 i
 *>  33.0.0.0          0.0.0.0                  0         32768 i
 *>  44.0.0.0          0.0.0.0                  0         32768 i
 *>  192.0.2.254/24    0.0.0.0                  0         32768 i
 *>  192.0.2.50        0.0.0.0                  0         32768 i
```

PE 1

```
PE1# show ip bgp all

For address family: IPv4 Unicast


For address family: VPNv4 Unicast

BGP table version is 46, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf red)
 *>  21.21.21.0/24    172.16.1.2               0             0 101 i
 *>  22.22.22.22/32   172.16.1.2               0             0 101 i
 *>  31.0.0.0         172.16.1.2               0             0 101 i
 *>  33.0.0.0         172.16.1.2               0             0 101 i
 *>  44.0.0.0         172.16.1.2               0             0 101 i
 *>  192.0.2.254/24   172.16.1.2               0             0 101 i
 *>  192.0.2.50       172.16.1.2               0             0 101 i
```

PE 4

Note The status is Withdraw for the exist-map EXIST-2 because the condition for advertisement has not been met.

```
PE4# show ip bgp all

For address family: VPNv4 Unicast

BGP table version is 82, local router ID is 10.0.0.4

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf red)
 *>i 21.21.21.0/24    10.0.0.1                 0    100      0 101 i
 *>i 22.22.22.22/32   10.0.0.1                 0    100      0 101 i
 *>i 31.0.0.0         10.0.0.1                 0    100      0 101 i
 *>i 33.0.0.0         10.0.0.1                 0    100      0 101 i
 *>i 44.0.0.0        10.0.0.1                0    100      0 101 I     <- missing 41.0.0.0/8

 *>i 192.0.2.254/24   10.0.0.1                 0    100      0 101 i
 *>i 192.0.2.50       10.0.0.1                 0    100      0 101 i
```

```
PE4# show ip bgp vpnv4 all neighbors 198.51.100.3
…
…
 For address family: VPNv4 Unicast
  Translates address family IPv4 Unicast for VRF red
  Session: 198.51.100.3
  BGP table version 48, neighbor version 48/0
  Output queue size : 0
  Index 3, Advertise bit 0
  3 update-group member
  Condition-map EXIST-1, Advertise-map ADV-1, status: Advertise
  Condition-map EXIST-2, Advertise-map ADV-2, status: Withdraw
  Condition-map EXIST-3, Advertise-map ADV-3, status: Advertise
  Condition-map EXIST-4, Advertise-map ADV-4, status: Advertise
  Slow-peer detection is disabled
  …
…
PE4#


PE4# show ip bgp vpnv4 all update-group

…
…
BGP version 4 update-group 3, external, Address Family: VPNv4 Unicast
  BGP Update version : 48/0, messages 0
  Condition-map EXIST-1, Advertise-map ADV-1, status: Advertise
  Condition-map EXIST-2, Advertise-map ADV-2, status: Withdraw
  Condition-map EXIST-3, Advertise-map ADV-3, status: Advertise
  Condition-map EXIST-4, Advertise-map ADV-4, status: Advertise
  Topology: red, highest version: 47, tail marker: 47
  Format state: Current working (OK, last not in list)
                Refresh blocked (not in list, last not in list)
  Update messages formatted 4, replicated 4, current 0, refresh 0, limit 1000
  Number of NLRIs in the update sent: max 3, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 1 member:
   198.51.100.3
```

CE 104

Note Prefix 44.0.0.0 is missing as 41.0.0.0/8 does not appear in PE 4 to trigger the advertisement to CE 104. The state is Withdraw.

```
CE104# show ip bgp all

For address family: IPv4 Unicast

BGP table version is 45, local router ID is 198.51.100.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  21.21.21.0/24    104.0.0.1                0      65000 101     i
 *>  22.22.22.22/32   104.0.0.1                0      65000 101     i
 *>  31.0.0.0         104.0.0.1                0      65000 101     i
```

```
*>  33.0.0.0          104.0.0.1             0      65000 101     i
*>  192.0.2.254/24    104.0.0.1             0      65000 101     i
*>  192.0.2.50        104.0.0.1             0      65000 101     i
```

Additional References for BGP VRF-Aware Conditional Advertisement

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP VRF-Aware Conditional Advertisement

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 140: Feature Information for BGP VRF-Aware Conditional Advertisement

Feature Name	Releases	Feature Information
BGP VRF-Aware Conditional Advertisement		The Border Gateway Protocol (BGP) VRF-Aware Conditional Advertisement feature provides additional control of the advertisement of routes and extends this control to within a virtual routing and forwarding (VRF) instance.

CHAPTER 111

BGP—Selective Route Download

The BGP—Selective Route Download feature allows a network administrator to selectively download some or none of the BGP routes into the Routing Information Base (RIB). The primary application for this feature is to suppress the unnecessary downloading of certain BGP routes to the RIB or Forwarding Information Base (FIB) on a dedicated route reflector, which propagates BGP updates without carrying transit traffic. The feature thereby helps to maximize resources available and to improve routing scalability and convergence on the dedicated route reflector.

Information About BGP—Selective Route Download

Dedicated Route Reflector Does Not Need All Routes

The role of a dedicated route reflector (RR) is to propagate BGP updates without participating in the actual forwarding of transit traffic. That means the RR does not need to have all BGP routes downloaded into its RIB or FIB. It is beneficial for the RR to preserve its resources by not processing and storing those routes.

By default, BGP routes are downloaded to the RIB. To save resources on a dedicated route reflector, such downloading can be reduced or prevented by configuring a table map. A table map is so named because it controls what is put into the BGP routing table.

A table map references a route map, in this context to control the downloading of routes. A table map can be used in other features, such as the BGP Policy Accounting Output Interface Accounting feature.

It is important to understand the use of the **filter** keyword in the **table-map** command.

- When the **table-map** command is used *without* the **filter** keyword, the route map referenced in the **table-map** command is used to set certain properties (such as the traffic index) of the routes for installation into the RIB. The route is always downloaded, regardless of whether it is permitted or denied by the route map.

- When the **table-map** command is used *with* the **filter** keyword, the route map referenced is also used to control whether a BGP route is to be downloaded to the RIB (hence the filter). A BGP route is not downloaded to the RIB if it is denied by the route map.

Note that the Selective Route Download feature is not applicable to Multiprotocol Label Switching (MPLS) Layer 3 VPN because the route download is already automatically suppressed on a route reflector.

Benefits of Selective Route Download

The BGP—Selective Route Download feature allows a network administrator to selectively download some or none of the BGP routes into the Routing Information Base (RIB). The primary application for this feature is to suppress the unnecessary downloading of certain BGP routes to the RIB or Forwarding Information Base (FIB) on a dedicated route reflector, which propagates BGP updates without carrying transit traffic. The feature thereby helps to maximize resources available and to improve routing scalability and convergence on the dedicated route reflector.

How to Selectively Download BGP Routes

Suppressing the Downloading of All BGP Routes on a Dedicated RR

Perform this task on a dedicated route reflector (RR) to prevent all BGP routes from being downloaded to the RIB, and thereby save resources.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *route-map-name* **deny** [*sequence-number*]
4. **exit**
5. **router bgp** *as-number*
6. **address-family ipv4 unicast**
7. **table-map** *route-map-name* **filter**
8. **end**
9. **clear ip bgp ipv4 unicast table-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**route-map** *route-map-name* **deny** [*sequence-number*] **Example:** `Router(config)# route-map bgp-to-rib deny 10`	Enters route map configuration mode to configure a route map. • In this example, the route map named bgp-to-rib denies all routes.
Step 4	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 5	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Enters router configuration mode and creates a BGP routing process.
Step 6	**address-family ipv4 unicast** **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address-family-specific configurations.
Step 7	**table-map** *route-map-name* **filter** **Example:** `Router(config-router-af)# table-map bgp-to-rib filter`	Specifies a route map that filters what goes into the BGP routing table (the Routing Information Base [RIB]). • The routes that are permitted by the route map are downloaded into the RIB. • The routes that are denied by the route map are filtered from (not downloaded into) the RIB.
Step 8	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 9	**clear ip bgp ipv4 unicast table-map** **Example:** `Router# clear ip bgp ipv4 unicast table-map`	Reloads the BGP RIB after the table map or the route map is configured or changed in order to put the changes into effect.

Selectively Downloading BGP Routes on a Dedicated RR

Perform this task on a dedicated route reflector (RR) to selectively download BGP routes to the RIB. When the externally connected routes are carried in BGP, it is necessary to download these routes to the RIB for next hop resolution on the RR. One scalable approach to accomplish the selective route download is to use a BGP community to identify the externally connected routes. That is, attach a designated BGP community during the redistribution of the externally connected routes on the ASBRs, and then on the RR, filter the route

download based on the BGP community. This task illustrates the configuration of the RR using a route map that matches on a community list to control which routes are downloaded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip community-list** *standard-list-number* **permit** *AA:NN*
4. **route-map** *route-map-name* **permit** [*sequence-number*]
5. **match community** *standard-list-number*
6. **exit**
7. **router bgp** *as-number*
8. **address-family ipv4 unicast**
9. **table-map** *route-map-name* **filter**
10. **end**
11. **clear ip bgp ipv4 unicast table-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip community-list** *standard-list-number* **permit** *AA:NN* **Example:** `Router(config)# ip community-list 100 permit 65510:100`	Creates a standard community list and specifies an autonomous system and network number allowed in the community list.
Step 4	**route-map** *route-map-name* **permit** [*sequence-number*] **Example:** `Router(config)# route-map bgp-to-rib permit 10`	Enters route-map configuration mode to configure a route map. • The route map named bgp-to-rib permits routes that match the community list identified in the next step.
Step 5	**match community** *standard-list-number* **Example:** `Router(config-route-map)# match community 100`	Matches on routes that are permitted by community list 100.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 7	**router bgp** *as-number* **Example:** `Router(config)# router bgp 65510`	Enters router configuration mode and creates a BGP routing process.
Step 8	**address-family ipv4 unicast** **Example:** `Router(config-router)# address-family ipv4 unicast`	Enters address family configuration mode to configure BGP peers to accept address family-specific configurations.
Step 9	**table-map** *route-map-name* **filter** **Example:** `Router(config-router-af)# table-map bgp-to-rib filter`	Specifies a route map that filters what goes into the BGP routing table (the Routing Information Base [RIB]).
Step 10	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.
Step 11	**clear ip bgp ipv4 unicast table-map** **Example:** `Router# clear ip bgp ipv4 unicast table-map`	Reloads the BGP RIB after the table map or the route map is configured or changed in order to put the changes into effect.

Configuration Examples for BGP—Selective Route Download

Examples: Selective Route Download

The role of a dedicated route reflector (RR) is to propagate BGP updates without participating in the actual forwarding of transit traffic. In some cases, the dedicated RR may need only selected routes downloaded; in some cases it may not need any routes downloaded.

It is likely that the dedicated RR would have the overload bit set if the IS-IS routing protocol is being used, or an OSPF stub router would be configured if OSPF is being used.

Example: Next Hop is Loopback Address—Filter All Routes From Being Downloaded

In this example, the ASBRs are configured with the **next-hop-self** command for iBGP sessions. (That configuration is not shown). The next hops of the BGP routes advertised to iBGP sessions are the loopback addresses carried in the IGP (either OSPF or IS-IS). There is no need to download any BGP routes to the RIB. The following configuration on the dedicated RR suppresses the downloading of all BGP routes because the **table map** command includes the **filter** keyword, and the route map that the table map references denies all routes.

```
route-map bgp-to-rib deny 10
!
router bgp 65000
 address-family ipv6 unicast
 table-map bgp-to-rib filter
```

Example: Redistribution of Connected Routes in IGP—Filter All Routes From Being Downloaded

In this example, the next hops of the BGP routes are resolved on the externally connected routes, which are carried in an IGP, such as OSPF or IS-IS, via a prefix-list-based selective redistribution of the connected routes. The routes are received from iBGP.

Although the scenario is different from the preceding example, the configuration is the same. The following configuration on the dedicated RR suppresses the downloading of all BGP routes because the **table map** command includes the **filter** keyword, and the route map that the table map references denies all routes.

```
route-map bgp-to-rib deny 10
!
router bgp 65000
 address-family ipv6 unicast
 table-map bgp-to-rib filter
```

Example: Redistribution of Connected Routes in BGP—Selectively Filter Routes From Being Downloaded

When the externally connected routes are carried in BGP, it is necessary to download these routes to the RIB, where the nexthop resolution on the RR can be calculated. One scalable way to achieve the selective route download is to use a BGP community on the ASBR to identify these externally connected routes. That is, on the border routers, attach a designated BGP community during the redistribution of the externally connected routes, and then on the RR, filter the route download based on the BGP community. The following shows the configuration on the ASBR and the configuration on the RR.

ASBR Configuration

```
router bgp 65510
 address-family ipv4 unicast
 redistribute connected route-map connected-to-bgp
!
route-map connected-to-bgp permit 10
 match ip address prefix-list extend-connected
 set community 65510:100
!
ip prefix-list extend-connected permit 192.168.1.1/30
```

RR Configuration

```
ip community-list 100 permit 65510:100
!
route-map bgp-to-rib permit 10
 match community 100
!
router bgp 65510
 address-family ipv4 unicast
 table-map bgp-to-rib filter
```

Additional References for Selective Route Download

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Commands List, All Releases
BGP Commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Selective Route Download

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 141: Feature Information for Selective Route Download

Feature Name	Releases	Feature Information
Selective Route Download	Cisco IOS XE Release 2.3S	The BGP—Selective Route Download feature allows a network administrator to selectively download some or none of the BGP routes into the Routing Information Base (RIB). The primary application for this feature is to suppress the unnecessary downloading of certain BGP routes to the RIB or Forwarding Information Base (FIB) on a dedicated route reflector, which propagates BGP updates without carrying transit traffic. The feature thereby helps to maximize resources available and to improve routing scalability and convergence on the dedicated route reflector. The following command was modified: • **table-map**

CHAPTER 112

BGP—Support for iBGP Local-AS

Prior to the BGP—Support for iBGP Local-AS feature, the **neighbor local-as** command was used on a BGP speaker to change the AS negotiated for an eBGP neighbor and to modify the AS_PATH sent and/or received. The **neighbor local-as** command can now be used to do the same on an iBGP session. AS negotiation creates an iBGP session and we enable sending iBGP attributes (LOCAL_PREF, ORIGINATOR_ID, and CLUSTER_LIST) over it, and accept this attributes when received from this session. This functionality is useful when merging two autonomous systems into one.

Restrictions for Support for iBGP Local-AS

- This feature is not supported for a peer that belongs to a confederation.
- Nonlocal-AS iBGP neighbors that are in a single AS are put into a separate update group from iBGP neighbors that are configured with the iBGP Local-AS feature.
- Two iBGP neighbors that are in two different autonomous systems and that are configured as iBGP Local-AS neighbors are put into separate update groups.

Information About Support for iBGP Local-AS

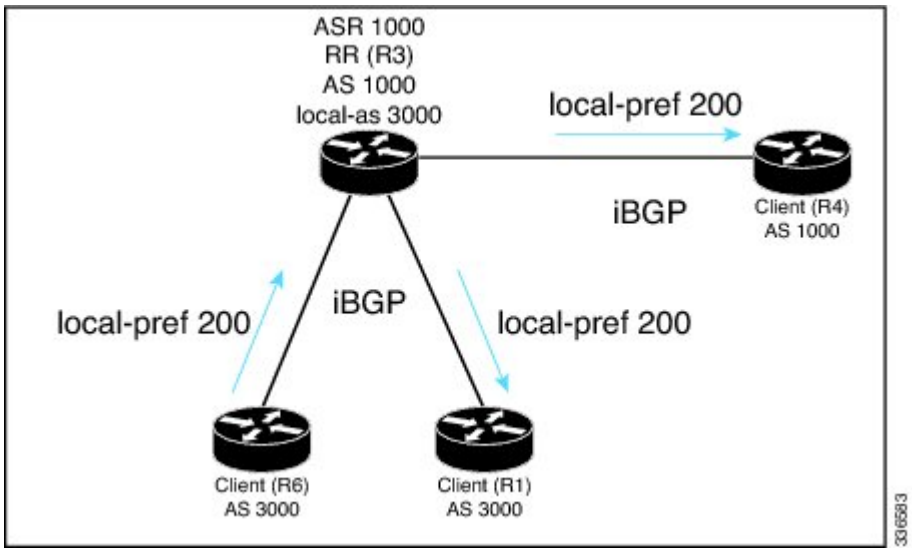
Support for iBGP Local-AS

Prior to the Support for iBGP Local-AS feature, when a peer (or peer group) was configured with the **neighbor local-as** command and the **neighbor remote-as** command that specified the same AS number, the session would be negotiated as an iBGP session (this happens when the advertised ASes in both OPEN messages are the same). However, updates were propagated as in an eBGP session (LOCAL_PREF, ORIGINATOR_ID and CLUSTER_LIST were not propagated), and could cause errors if they were received via this session. Thus, iBGP local-AS was not fully supported.

The Support for iBGP Local-AS feature means all those iBGP attributes are propogated. Additionally, as in any iBGP session, the AS is not prepended in AS_PATH attribute when advertising routes to an iBGP local-as session.

The figure below illustrates a scenario where this feature is being used to facilitate the merging of two autonomous systems. The route reflector R3 and R4 belong to AS 1000; R1 and R6 belong to AS 3000. The RR is configured with **neighbor local-as 3000** and **neighbor remote-as 3000** commands. Even though the routers belong to two different autonomous systems, attributes like the LOCAL_PREF are preserved in the updates from R6 to R4 and R6 to R1 (as show in the figure), and also in the updates from R4 to R1 and R4 to R6 (not shown in the figure).

Figure 126: Support for iBGP Local-AS to Preserve iBGP Policies Between Two Autonomous Systems



Benefits of iBGP Local-AS

This feature is used when merging two ISPs that have different autonomous system numbers. It is desirable to preserve attributes that are considered internal (LOCAL_PREF, ORIGINATOR_ID, and CLUSTER_LIST) in the routes that are being propogated to other autonomous system.

How to Configure iBGP Local-AS

Configuring iBGP Local-AS

Configure the iBGP Local-AS feature on a BGP speaker for a given neighbor when you want that session to behave as a full iBGP session. This configuration is typically performed on a route reflector, but not exclusively on it. In a route reflector you can optionally configure changing iBGP attributes sent to a neighbor via the command **allow-policy** (this command is not exclusive for this feature and can be used on any RR).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **router bgp** *autonomous-system-number*
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **remote-as** *as-number*
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **local-as** *as-number*
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **route-reflector-client**
10. **address-family vpnv4**
11. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **allow-policy**
12. **exit**
13. **address-family vpnv6**
14. **neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **allow-policy**
15. **end**
16. **show ip bgp vpnv4 all neighbors** {*ip-address* | *ipv6-address*} **policy**
17. **show ip bgp vpnv4 all update-group** *update-group*
18. **show ip bgp vpnv4 all neighbors** {*ip-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

	Command or Action	Purpose
Step 4	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 1000`	Enters router configuration mode to create or configure a BGP routing process.
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-router)# neighbor rr-client-ab peer-group`	(Optional) Identifies a peer group.
Step 6	**neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name* **Example:** `Device(config-router)# neighbor 192.168.3.3 peer-group rr-client-ab`	(Optional) Configures a BGP neighbor to be a member of a peer group.
Step 7	**neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **remote-as** *as-number* **Example:** `Device(config-router)# neighbor rr-client-ab remote-as 3000`	Identifies the AS of the neighbor or peer group.
Step 8	**neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **local-as** *as-number* **Example:** `Device(config-router)# neighbor rr-client-ab local-as 3000`	Configures the local-AS feature for the neighbor or peer group.
Step 9	**neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **route-reflector-client** **Example:** `Device(config-router)# neighbor rr-client-ab route-reflector-client`	Configures the local device to be a route reflector and configures the neighbor or peer group to be its client.
Step 10	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	(Optional) Places the router in VPNv4 address family configuration mode.
Step 11	**neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **allow-policy** **Example:** `Device(config-router-af)# neighbor rr-client-ab allow-policy`	(Optional) Allows the RR to be configured to change iBGP attributes for the specified neighbor or peer group.
Step 12	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and enters router configuration mode.

	Command or Action	Purpose
Step 13	**address-family vpnv6** **Example:** `Device(config-router)# address-family vpnv6`	(Optional) Places the router in VPNv6 address family configuration mode.
Step 14	**neighbor** {*ip-address* | *ipv6-address* | *peer-group*} **allow-policy** **Example:** `Device(config-router-af)# neighbor rr-client-ab allow-policy`	(Optional) Allows the RR to be configured to change iBGP attributes for the specified neighbor or peer group.
Step 15	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode, and enters privileged EXEC mode.
Step 16	**show ip bgp vpnv4 all neighbors** {*ip-address* | *ipv6-address*} **policy** **Example:** `Device# show ip bgp vpnv4 all neighbors 192.168.3.3 policy`	(Optional) Displays the locally configured policies of the neighbor. • The output includes the phrase "allow-policy" if the **neighbor allow-policy** command was configured for that neighbor.
Step 17	**show ip bgp vpnv4 all update-group** *update-group* **Example:** `Device# show ip bgp vpnv4 all update-group 2`	(Optional) Displays the information for the update group. • The output includes the phrase "Allow-policy" if the **neighbor allow-policy** command was configured for neighbors in the update group.
Step 18	**show ip bgp vpnv4 all neighbors** {*ip-address* | *ipv6-address*} **Example:** `Device# show ip bgp vpnv4 all neighbors 192.168.3.3`	(Optional) Displays information about the neighbor. • The output includes the remote AS and local AS, which will indicate the same AS number when the Support for iBGP Local-AS feature is configured.

Configuration Examples for iBGP Local-AS

Example: Configuring iBGP Local-AS

The example configures a route reflector (RR) in AS 4000 to treat BGP sessions with the peer group rr-client-2 in AS 2500 as iBGP sessions. That is, iBGP attributes (LOCAL_PREF, ORIGINATOR_ID, and CLUSTER_LIST) will not be dropped from routes in advertisements to and from the neighbors belonging to the peer group; the attributes will be passed unmodified. AS 2500 will not be prepended to the AS_PATH attribute in routes to or from the peer group.

Additionally, the **neighbor allow-policy** command configures that the network administrator can configure iBGP policies on the RR. That is, an outbound route map can be configured to change attributes that are sent to the downstream peers. In this example, the command is applied to VPNv4 and VPNv6 address families.

```
router bgp 4000
 neighbor rr-client-2 peer-group
 neighbor 192.168.1.1 peer-group rr-client-2
 neighbor 192.168.4.1 peer-group rr-client-2
 neighbor rr-client-2 remote-as 2500
 neighbor rr-client-2 local-as 2500
 neighbor rr-client-2 route-reflector-client
 address-family vpnv4
  neighbor rr-client-2 allow-policy
!
 address-family vpnv6
  neighbor rr-client-2 allow-policy
```

Additional References for Support for iBGP Local-AS

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Migration of autonomous systems	"BGP Support for Dual AS Configuration for Network AS Migrations" module in the *IP Routing: BGP Configuration Guide, Cisco IOS XE Release 3S*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP—Support for iBGP Local-AS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 142: Feature Information for BGP—Support for iBGP Local-AS

Feature Name	Releases	Feature Information
BGP—Support for iBGP Local-AS		Prior to the BGP—Support for Local-AS feature, the **neighbor local-as** command was used on a route reflector to customize AS_PATH attributes for routes received from an eBGP neighbor. The **neighbor local-as** command can now be used to enable the sending of the iBGP attributes (LOCAL_PREF, ORIGINATOR_ID, CLUSTER_ID, and CLUSTER_LIST) over an iBGP local-AS session. This functionality is useful when merging two autonomous systems, when it is advantageous to keep the iBGP attributes in routes. Prior to the BGP—Support for iBGP Local-AS feature, the RR should not have been configured to change iBGP attributes. With the introduction of this feature, the RR can be configured to change iBGP attributes, providing more flexibility. The following command was introduced: • **neighbor allow-policy** The following commands were modified: • **neighbor local-as** • **show ip bgp vpnv4**

CHAPTER 113

eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

The eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6) feature allows you to configure multipath load sharing among native IPv4 and IPv6 external Border Gateway Protocol (eBGP) and internal BGP (iBGP) paths for improved load balancing in deployments. This module explains the feature and how to configure it.

Information About eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

eiBGP Multipath for Non-VRF Interfaces Overview

The Border Gateway Protocol (BGP) path-selection algorithm prefers external BGP (eBGP) paths over internal BGP (iBGP) paths. With the eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6) feature, this algorithm is modified to allow multipath load sharing among native IPv4 and IPv6 eBGP and iBGP paths. Prior to the eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6) feature, this functionality was only available on VPN routing and forwarding (VRF) instances. With this feature, the functionality is extended to non-VRF interfaces. The **maximum-paths** command allows you to configure BGP to install multiple paths in the Routing Information Base (RIB) for multipath load sharing. The BGP best path algorithm selects a single multipath as the best path and advertises the path to BGP peers. Other multipaths are inserted into both the BGP table and the RIB, and these multipaths are used by Cisco Express Forwarding to perform load balancing, which is performed either on a per-packet basis or on a per-source or per-destination basis.

This feature can be configured on a customer provider edge (PE) device. However, the feature should be configured only on one PE device at the customer site. If this feature is configured on more than one PE device, some parts of the traffic may loop between the PE devices at the customer site. Therefore, it is important to set up the feature appropriately to avoid traffic loops. This feature is enabled by default.

How to Configure eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

Enabling IPv4/IPv6 Multipaths for Non-VRF Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. Enter one of the following:
   - **address-family ipv4 unicast**
   - **address-family ipv6 unicast**
5. **maximum-paths eibgp** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 64496`	Enters router configuration mode to create or configure a Border Gateway Protocol (BGP) routing process.

Step 4	Enter one of the following:
	• **address-family ipv4 unicast**
	• **address-family ipv6 unicast**
	Example:
	`Device(config-router)# address-family ipv4 unicast`
	`Device(config-router)# address-family ipv6 unicast`
		Enters IPv4 or IPv6 address family configuration mode.

Step 5	**maximum-paths eibgp** *number* **Example:** `Device(config-router-af)# maximum-paths eibgp 3`	Forwards packets over multiple external BGP (eBGP) and internal BGP (iBGP) paths.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

Example: Enabling IPv4/IPv6 Multipaths in Non-VRF Interfaces

The following example shows how to enable IPv4 multipaths on non-VRF interfaces.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 64496
Device(config-router)# address-family ipv4 unicast
Device(config-router-af)# maximum-paths eibgp 4
Device(config-router-af)# end
```

The following example shows how to enable IPv6 multipaths on non-VRF interfaces.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 64497
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# maximum-paths eibgp 4
Device(config-router-af)# end
```

Feature Information for eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 143: Feature Information for eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)

Feature Name	Releases	Feature Information
eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6)		The eiBGP Multipath for Non-VRF Interfaces (IPv4/IPv6) feature allows you to configure multipath load sharing among native IPv4 and IPv6 external Border Gateway Protocol (eBGP) and internal BGP (iBGP) paths for improved load balancing in deployments. The following command was modified: **maximum-paths eibgp**.

CHAPTER 114

L3VPN iBGP PE-CE

The L3VPN iBGP PE-CE feature enables the provider edge (PE) and customer edge (CE) devices to exchange Border Gateway Protocol (BGP) routing information by peering as iBGP instead of as external BGP peering between the PE and CE.

Restrictions for L3VPN iBGP PE-CE

We recommend not using the soft-reconfiguration inbound or BGP soft-reconfig-backup feature with the iBGP PE CE.

Information About L3VPN iBGP PE-CE

L3VPN iBGP PE-CE

When BGP is used as the provider edge (PE) or customer edge (CE) routing protocol, the peering sessions are configured as an external peering between the VPN provider autonomous system (AS) and the customer network autonomous system. The L3VPN iBGP PE-CE feature enables the PE and CE devices to exchange Border Gateway Protocol (BGP) routing information by peering as internal Border Gateway Protocol (iBGP) instead of the widely used external BGP peering between the PE and the CE. This mechanism applies at each PE device where a VRF-based CE is configured as iBGP. This eliminates the need for service providers (SPs) to configure autonomous system override for the CE. With this feature enabled, there is no need to configure the virtual private network (VPN) sites using different autonomous systems.

The introduction of the **neighbor internal-vpn-client** command enables PE devices to make an entire VPN cloud act like an internal VPN client to the CE devices. These CE devices are connected internally to the VPN cloud through the iBGP PE-CE connection inside the VRF. After this connection is established, the PE device encapsulates the CE-learned path into an attribute called ATTR_SET and carries it in the iBGP-sourced path throughout the VPN core to the remote PE device. At the remote PE device, this attribute is assigned with

individual attributes and the source CE path is extracted and sent to the remote CE devices. ATTR_SET is an optional transitive attribute that carries a set of BGP path attributes. It can include any BGP attribute that can occur in a BGP update message as received from the source CE device.

How to Configure L3VPN iBGP PE-CE

Configuring L3VPN iBGP PE-CE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4 vrf** *name*
5. **neighbor** *ip-address* **internal-vpn-client**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 100`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4 vrf** *name* **Example:** `Device(config-router)# address-family ipv4 vrf blue`	Enters address family configuration mode and configures VPN routing and forwarding.
Step 5	**neighbor** *ip-address* **internal-vpn-client** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 internal-vpn-client`	Defines a neighboring device with which to exchange routing information. The **neighbor internal-vpn-client** command stacks the iBGP-CE neighbor path in the VPN attribute set .

Configuration Examples for L3VPN iBGP PE-CE

Example: Configuring L3VPN iBGP PE-CE

The following example shows how to configure L3VPN iBGP PE-CE:

```
Device# enable
Device(config)# configure terminal
Device(config)# router bgp 100
Device(config-router)# address-family ipv4 vrf blue
Device(config-router-af)# neighbor 10.0.0.1 internal-vpn-client
```

Additional References for L3VPN iBGP PE-CE

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for L3VPN iBGP PE-CE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 144: Feature Information for L3VPN iBGP PE-CE

Feature Name	Releases	Feature Information
L3VPN iBGP PE-CE		The L3VPN iBGP PE-CE feature enables the provider edge (PE) and customer edge (CE) devices to exchange Border Gateway Protocol (BGP) routing information by peering as iBGP instead of as external BGP between the PE and CE. The **neighbor internal-vpn-client** command was introduced.

CHAPTER 115

BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B

Border Gateway Protocol (BGP) nonstop routing (NSR) provides support for NSR and nonstop forwarding (NSF) in the event of a switchover from an active to a standby Route Processor (RP). BGP NSR supports provider-edge-to-customer-edge (PE-CE) connections for IPv4 and IPv6 address families and also for Internal BGP (IBGP) peers at the PE device for IPv4, IPv6, VPN version 4 (VPNv4), and VPN version 6 (VPNv6) address families. The BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B feature provides support for NSR at the autonomous system boundary routers (ASBRs) in Multiprotocol Label Switching (MPLS) Inter-Autonomous System (Inter-AS) Option B deployments for both VPNv4 and VPNv6 address families.

This module describes how to enable BGP NSR support at ASBRs in Inter-AS Option B for VPNv4 and VPNv6 address families.

Restrictions for BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B

- If a peer is activated under an address family for which nonstop routing (NSR) is not supported (for example, multicast distribution tree [MDT]), and if the address family topology is tied to the same session as other address family topologies for which NSR is supported (for example, VPN version 4 [VPNv4]), then NSR will not be supported for that peer-established session. NSR cannot be supported for a session if the session establishment involves activating the peer in an address family for which NSR is not supported. As a workaround, you can create a multisession and activate the nonsupported topology as part of a new session.
- NSR can be configured only on a per-neighbor basis.

- There can be some performance and memory impact as a result of enabling BGP NSR support at autonomous system boundary routers (ASBRs) in Inter-AS Option B.

Information About BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B

Overview of BGP NSR

Border Gateway Protocol (BGP) nonstop routing (NSR) with stateful switchover (SSO) provides a high availability (HA) solution to service providers whose provider edge (PE) routers engage in External BGP (EBGP) peering relationships with customer edge (CE) routers that do not support BGP graceful restart (GR). BGP NSR works with SSO to synchronize BGP state information between the active and standby Route Processors (RPs). SSO minimizes the amount of time for which a network is unavailable to users following a switchover.

BGP NSR with SSO is supported in BGP peer, BGP peer group, and BGP session template configurations.

To configure support for BGP NSR with SSO in BGP peer and BGP peer group configurations, use the **neighbor ha-mode sso** command in address family configuration mode for IPv4 virtual routing and forwarding (VRF) address family BGP peer sessions. To include support for Cisco BGP NSR with SSO in a BGP session template, use the **ha-mode sso** command in session-template configuration mode.

Inter-Autonomous Systems

BGP autonomous systems (ASs) are used to divide global external networks into individual routing domains where local routing policies are applied. Separate BGP ASs dynamically exchange routing information through External BGP (EBGP) peering sessions. BGP peers within the same AS exchange routing information through Internal BGP (IBGP) peering sessions.

When multiple sites of a VPN are connected to different ASs, Inter-Autonomous System (Inter-AS) deployments are useful for providing VPN services between different ASs. In this scenario, provider edge (PE) routers attached to the VPN cannot maintain IBGP connections with each other or with a common route reflector (RR). EBGP is used to distribute VPN-IPv4/IPv6 addresses. RFC 2547bis presents the following Inter-AS VPN solutions:

- Virtual routing and forwarding (VRF)-to-VRF connections at autonomous system boundary routers (ASBRs)—PEs act as ASBRs of their ASs. The ASBRs are directly connected and manage VPN routes between them through multiple subinterfaces. The ASBRs associate each such subinterface with a VRF and use EBGP to distribute unlabeled IPv4 addresses to each other. This solution is also called "Inter-AS Option A." Inter-AS Option A provides IP-based forwarding between the ASBRs connecting the different ASs; however, it also requires a single BGP session for each VPN connection. Inter-AS Option A is easy to implement, but it has limited scalability.
- EBGP redistribution of labeled VPN-IPv4 routes—Neighboring ASBRs use Multiprotocol External BGP (MP-EBGP) to exchange labeled VPN-IPv4 routes that the ASBRs obtain from PEs in their respective ASs. PE routers use IBGP to redistribute labeled VPN-IPv4 routes either to an ASBR or to an RR of which an ASBR is a client. This solution is also called "Inter-AS Option B." Inter-AS Option B provides Multiprotocol Label Switching (MPLS)-based forwarding between the ASBRs connecting different ASs. Inter-AS Option B provides better scalability than Inter-AS Option A because Option B requires only one BGP session to exchange all VPN prefixes between the ASBRs.

- Multihop EBGP redistribution of labeled VPN-IPv4 routes—PEs exchange labeled VPN-IPv4 routes directly with each other through MP-EBGP without the participation of ASBRs. ASBRs advertise labeled IPv4 routes to PEs in their respective ASs through MP-IBGP. ASBRs neither maintain VPN-IPv4 routes nor advertise VPN-IPv4 routes to each other. This solution is also called "Inter-AS Option C."

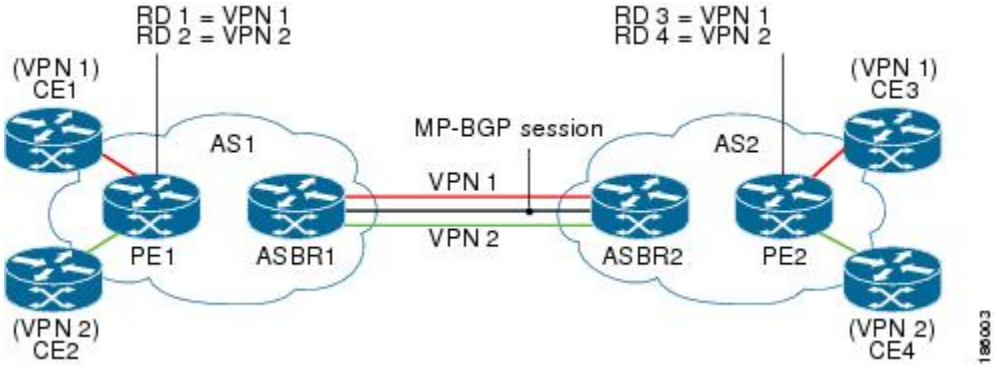
Overview of MPLS VPNv4 and VPNv6 Inter-AS Option B

In the Inter-Autonomous System (Inter-AS) Option B solution, two autonomous system border routers (ASBRs) use Multiprotocol External BGP (MP-EBGP) to exchange labeled VPN-IPv4 routes that they obtain from the provider edge (PEs) devices in their respective ASs. Multiprotocol Label Switching (MPLS)-based forwarding is used between the ASBRs. If a failure is encountered at an ASBR, routing and forwarding is impacted in the absence of nonstop routing (NSR) or graceful restart (GR). NSR provides the ability to preserve the routing state to a redundant Route Processor (RP), which can take over the functionality of the active RP in the event of a failover. In conjunction with nonstop forwarding (NSF), the routing and forwarding states can remain unimpacted during a failover.

The figure below illustrates two ASs, AS1 and AS2, each containing customer edge (CE) routers that belong to different VPNs. Each PE tracks which route distinguisher (RD) corresponds to which VPN, thus controlling the traffic that belongs to each VPN.

- Customer edge 1 (CE1) and CE3 belong to VPN 1.
- CE2 and CE4 belong to VPN 2.
- Provider edge 1 (PE1) uses route distinguisher 1 (RD 1) for VPN 1 (VRF 1) and RD 2 for VPN 2 (VRF 2).
- PE2 uses RD 3 for VPN 1 (VRF 1) and RD 4 for VPN 2 (VRF 2).

Figure 127: Flow of Routes in Inter-AS Option B



In an Inter-AS Option B scenario like the one in the figure above, the routes are carried across an AS boundary from ASBR1 to ASBR2 over an MP-EBGP session.

In Inter-AS Option B, the routes are advertised as follows:

1. PEs in AS1 advertise labeled VPN-IPv4 routes to either the ASBR of AS1 or the route reflector (RR) of the ASBR through Multiprotocol Internal BGP (MP-IBGP).
2. The ASBR of AS1 advertises labeled VPN-IPv4 routes to the ASBR of AS2 through MP-EBGP.
3. The ASBR of AS2 advertises labeled VPN-IPv4 routes to either the PEs in AS2 or the RR of the PEs through MP-IBGP.

The ASBRs must perform special processing on the labeled VPN-IPv4 routes, which is also called the ASBR extension method.

How to Configure BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B

Configuring an ASBR to Enable BGP NSR Support in Inter-AS Option B

Border Gateway Protocol (BGP) nonstop routing (NSR) support at autonomous system boundary router (ASBR) in Inter-Autonomous System (Inter-AS) Option B can be configured in the same way that BGP NSR is configured for Multiprotocol Internal BGP (MP-IBGP) peers at the provider edge (PE). The configuration is performed in global router mode, on a per-neighbor basis. The NSR support is applied to all address families under which the neighbor has been activated (provided the neighbor is not activated under a nonsupported address family). If a neighbor is activated under an unsupported address family, that topology must be made to be part of a different session using multisession.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **ha-mode sso**
6. **address-family** {**vpnv4** | **vpnv6**} [**multicast** | **unicast**]
7. **neighbor** *ip-address* **activate**
8. **end**
9. **show ip bgp vpnv4 all sso summary**
10. **show ip bgp vpnv4 neighbors** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 400`	Enters router configuration mode for the specified routing process.
Step 4	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:**	Specifies the AS of the neighbor.

	Command or Action	Purpose
	`Device(config-router)# neighbor 192.168.1.1 remote-as 4000`	
Step 5	**neighbor** *ip-address* **ha-mode sso** **Example:** `Device(config-router)# neighbor 192.168.1.1 ha-mode sso`	Configures a BGP neighbor to support BGP NSR with stateful switchover (SSO).
Step 6	**address-family** {**vpnv4** | **vpnv6**} [**multicast** | **unicast**] **Example:** `Device(config-router)# address-family vpnv4 unicast`	Enters address family configuration mode for configuring routing sessions that use standard VPNv4 or VPNv6 address prefixes.
Step 7	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 activate`	Activates the specified peer.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 9	**show ip bgp vpnv4 all sso summary** **Example:** `Device# show ip bgp vpnv4 all sso summary`	Displays information about BGP peers that support BGP NSR with SSO.
Step 10	**show ip bgp vpnv4 neighbors** *ip-address* **Example:** `Device# show ip bgp vpnv4 neighbors 192.168.1.1`	Displays information about BGP and TCP connections to neighbors.

Configuration Examples for BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B

Example: Configuring an ASBR to Enable BGP NSR Support in Inter-AS Option B

Configuring an ASBR to Be NSR-Capable at the VPNv4 Address Family Level

```
router bgp 200
  neighbor 192.168.1.1 remote-as 200
  neighbor 192.168.1.1 ha-mode sso
  address-family vpnv4 unicast
    neighbor 192.168.1.1 activate
```

Configuring an ASBR to Be NSR-Capable at the VPNv6 Address Family Level

```
router bgp 300
  neighbor 192.168.1.10 remote-as 300
  neighbor 192.168.1.10 ha-mode sso
  address-family vpnv6 multicast
    neighbor 192.168.1.10 activate
```

To verify that an ASBR is NSR-capable, check the **show ip bgp vpnv4 neighbors** command output for the Stateful switchover support enabled field.

```
ASBR# show ip bgp vpnv4 neighbors 192.168.1.10

BGP neighbor is 192.168.1.10,  vrf A,  remote AS 200, external link
  BGP version 4, remote router ID 192.168.1.10
  BGP state = Established, up for 00:16:01
  Last read 00:00:04, last write 00:00:35, hold time is 180, keepalive interval is 60 seconds

  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: YES for session 1
```

Additional References for BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 145: Feature Information for BGP NSR Support for Inter-AS Option B

Feature Name	Releases	Feature Information
BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B	Cisco IOS XE Release 3.10S	The BGP NSR Support for MPLS VPNv4 and VPNv6 Inter-AS Option B feature provides support for nonstop routing (NSR) at the autonomous system boundary routers (ASBR) in Inter-Autonomous System (Inter-AS) Option B deployments for both VPNv4 and VPNv6 address families. No commands were introduced or modified.

CHAPTER 116

BGP-RTC for Legacy PE

The BGP-Route Target Constrain (RTC) for Legacy PE feature helps to prevent the propagation of VPN Network Layer Reachability Information (NLRI) to a provider edge (PE) device that is not interested in the VPN. This feature builds an outbound filter used by a Boarder Gateway Protocol (BGP) speaker to decide which routes to pass to its peer and propagates route target (RT) reachability information between internal BGP (iBGP) meshes.

Prerequisites for BGP-RTC for Legacy PE

Before you configure the BGP-RTC for Legacy PE feature, you must configure the RT filter unicast address family type. For more information, see "Configuring BGP: RT Constrained Route Distribution" module in the *IP Routing: BGP Configuration Guide*.

Information About BGP-RTC for Legacy PE

Overview of BGP-RTC for Legacy PE

The BGP—RTC for Legacy PE feature makes use of VPN unicast route exchange from the legacy provider edge (PE) devices to a new Boarder Gateway Protocol (BGP) speaker (route reflector [RR]) to signal route target (RT) membership. The legacy PEs announce a set of special routes with mapped RTs to the RR along with a standard community. The presence of the community triggers the RR to extract the RTs and build RT membership information.

In scenarios where VPN membership is normal, this functionality helps reduce the scaling requirements on the PE devices and the RRs. The PE devices need not to spend resources for filtering out unwanted routes. The BGP peers that have common outbound policies are grouped under a single format group. Separate replication groups are used within a format group to separate BGP peers with its own peer-based policies. The Route Target Constrain (RTC)-capable peers are placed in separate format groups. Each RTC peers have

a separate replication group. When legacy RT is configured for a peer, then it must be treated the same way as the RTC peer except that there is no capability negotiation.

Legacy PE Support-PE Behavior

Each legacy Route Target Constrain (RTC) speaking neighbor is assigned a separate replication group. BGP checks the VPN table for any route with a reserved community value and uses it to create RTC network from the VPN prefix received from a legacy RTC peer with community values. The PE device uses the existing VPN advertisement mechanism to convey route target (RT) membership from the legacy provider edge (PE) devices. The route reflector (RR) processes advertisement mechanisms of RT membership information from legacy PE devices. RRs translate the legacy PE RT membership information to equivalent RTC Network Layer Reachability Information (NLRIs) to propagate to other RRs.

Legacy PE Support-RR Behavior

Route reflectors (RR) identify routes from legacy provider edge (PE) devices for retrieving route target (RT) membership information by the community value and filter VPN routes to legacy PE devices. RRs use the existing VPN advertisement mechanism to convey and process RT membership from the legacy PEs. The legacy PE RT membership information is translated into equivalent RT membership Network Layer Reachability Information (NLRI) from the client to propagate to other RRs. The RR then creates the route target filter list for each legacy client by collecting the entire set of route targets.

How to Configure BGP-RTC for Legacy PE

Configuring BGP-RTC for Legacy PE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family** {**vpnv4** | **vpnv6** } **unicast**
5. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **accept-route-legacy-rt**
6. **address-family rtfilter**
7. **end**
8. **show ip bgp vpnv4 all update-group** *update-group*
9. **show ip bgp vpnv4 all neighbors** {*ip-address* | *ipv6-address*}
10. **show ip bgp vpnv4 all peer-group**
11. **debug ip bgp all updates in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 1`	Configures a Boarder Gateway Protocol (BGP) routing process and enters router configuration mode.
Step 4	**address-family** {**vpnv4** | **vpnv6** } **unicast** **Example:** `Device(config-router)# address-family vpnv4 unicast`	Specifies the VPNv4 or VPNv6 address family and enters address family configuration mode.
Step 5	**neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **accept-route-legacy-rt** **Example:** `Device(config-router-af)# neighbor 10.0.0.1 accept-route-legacy-rt`	Configures the neighbor on the route reflector (RR) to treat the provider edge (PE) device as a legacy PE for the route target (RT) and accepts VPN routes tagged with the special community.
Step 6	**address-family rtfilter** **Example:** `Device(config-router-af)# address-family rtfilter`	Specifies the RT filter address family type.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 8	**show ip bgp vpnv4 all update-group** *update-group* **Example:** `Device# show ip bgp vpnv4 all update-group 2`	(Optional) Displays the information about neighbors in the update group.
Step 9	**show ip bgp vpnv4 all neighbors** {*ip-address* | *ipv6-address*} **Example:** `Device# show ip bgp vpnv4 all neighbors 192.168.3.3`	(Optional) Displays information about the BGP VPNv4 neighbor.
Step 10	**show ip bgp vpnv4 all peer-group** **Example:** `Device# show ip bgp vpnv4  all peer-group`	(Optional) Displays information about the peer groups.
Step 11	**debug ip bgp all updates in** **Example:** `Device# debug ip bgp all updates in`	(Optional) Displays BGP update messages.

Configuration Examples for BGP-RTC for Legacy PE

Example: BGP-RTC for Legacy PE

Configuration on the Route Reflector

The following example shows how to configure the neighbor on the route reflector (RR) to treat the provider edge (PE) device as a legacy PE for the route target (RT) and accept VPN routes tagged with the special community:

```
Device# configure terminal
Device(config)# router bgp 1
Device(config-router)# address-family vpnv4 unicast
Device(config-router-af)# neighbor 10.1.1.1 accept-route-legacy-rt
Device(config-router-af)# address-family rtfilter
Device(config-router-af)# exit address-family
```

Configuration on the Legacy PE

The following example shows how to create a route filter VRF and attach an export map that collects and carries all RTs locally configured on Layer 3 VPN virtual routing and forwarding (VRF):

```
ip vrf route-filter
 rd 55:1111
 export map SET_RT

route-map SET_RT permit 10
 match ip address prefix-list RT_NET1
 set community 4294901762 (0xFFFF0002)
 set extcommunity rt  255.220.0.0:12241 255.220.0.0:12242 additive
 set extcommunity rt  255.220.0.0:12243 255.220.0.0:12244 additive
 set extcommunity rt  255.220.0.0:12245 255.220.0.0:12246 additive
 set extcommunity rt  255.220.0.0:12247 255.220.0.0:12248 additive
 set extcommunity rt  255.220.0.0:12249 255.220.0.0:12250 additive
!
route-map SET_RT permit 20
 match ip address prefix-list RT_NET2
 set community 4294901762 (0xFFFF0002)
 set extcommunity rt  255.220.0.0:12251 255.220.0.0:12252 additive
 set extcommunity rt  255.220.0.0:12253 255.220.0.0:12254 additive
 set extcommunity rt  255.220.0.0:12255 additive
!

ip route vrf route-filter 5.5.5.5 255.255.255.255 Null0 – (matching prefix-set RT_NET1)
ip route vrf route-filter 6.6.6.6 255.255.255.255 Null0 –(matching prefix-set RT_NET2)

route-map LEG_PE permit 10
 match ip address prefix-list RT_NET1 RT_NET2
 set community no-advertise additive
```

The following example shows how to apply the route map to a VPNv4 neighbor:

```
router bgp 55
 address-family vpnv4 unicast
 neighbor x.x.x.x  route-map LEG_PE out
```

The following example shows how to source a static route into a Boarder Gateway Protocol (BGP) network using a network statement:

```
router bgp 55
 address-family ipv4 vrf route-filter
 network 5.5.5.5 mask 255.255.255.255
 network 6.6.6.6 mask 255.255.255.255
```

Additional References for BGP-RTC for Legacy PE

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
Configuring BGP: RT Constrained Route Distribution	"Configuring BGP: RT Constrained Route Distribution" module in the *IP Routing: BGP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 4684	*Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP-RTC for Legacy PE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 146: Feature Information for BGP-RTC for Legacy PE

Feature Name	Releases	Feature Information
BGP-RTC for Legacy PE		The BGP-RTC for Legacy PE feature helps to prevent the propagation of VPN Network Layer Reachability Information (NLRI) to a provider edge (PE) device that is not interested in the VPN. This feature builds an outbound filter used by a Boarder Gateway Protocol (BGP) speaker to decide which routes to pass to its peer and propagates route target (RT) reachability information between internal BGP (iBGP) meshes. The **neighbor accept-route-legacy-rt** command was introduced.

CHAPTER 117

BGP PBB EVPN Route Reflector Support

The BGP PBB EVPN Route Reflector Support feature provides Boarder Gateway Protocol (BGP) route reflector functionality for Ethernet VPN (EVPN) and provider backbone bridging (PBB) EVPN of Layer 2 VPN address family. EVPN enables customer MAC addresses as routable addresses and distributes them in BGP to avoid any data plane MAC address learning over the Multiprotocol Label Switching (MPLS) core network. The route reflector is enhanced to store the received EVPN updates without configuring EVPN explicitly on the route reflector and then advertises these updates to other provider edge (PE) devices so that the PEs do not need to have a full mesh of BGP sessions.

Prerequisites for BGP PBB EVPN Route Reflector Support

- Before you configure the BGP PBB EVPN Route Reflector Support feature, you must configure the RT filter unicast address family type to support for EVPN address family. For more information, see the "Configuring BGP: RT Constrained Route Distribution" module in the *IP Routing: BGP Configuration Guide*.
- The EVPN Subsequent Address Family Identifier (SAFI) needs to be enabled globally before you enable it under the BGP neighbor.

Information About BGP PBB EVPN Route Reflector Support

EVPN Overview

Ethernet VPN (EVPN) allows Multiprotocol Label Switching (MPLS) networks to provide multipoint Layer 2 VPN (L2VPN) services.

In EVPN, the customer MAC addresses are learned in the data plane over links connecting customer devices (CE) to the provider edge (PE) devices. The MAC addresses are then distributed over the Multiprotocol Label

Switching (MPLS) core network using Boarder Gateway Protocol (BGP) with an MPLS label identifying the service instance. A single MPLS label per EVPN instance is sufficient as long as the receiving PE device performs a MAC lookup in the disposition path. Receiving PE devices inject these routable MAC addresses into their Layer 2 routing information base (RIB) and forwarding information base (FIB) along with their associated adjacencies.

EVPN defines a BGP Network Layer Reachability Information (NLRI) that advertises different route types and route attributes. The EVPN NLRI is carried in BGP using BGP multiprotocol extensions with an Address Family Identifier (AFI) and a Subsequent Address Family Identifier (SAFI). BGP drops unsupported route types and does not propagate them to neighbors.

BGP EVPN Autodiscovery Support on Route Reflector

By default, routes received from an internal BGP (iBGP) peer are not sent to another iBGP peer unless a full mesh configuration is formed between all Boarder Gateway Protocol (BGP) devices within an autonomous system (AS). Configuring a route reflector allows a device to advertise or reflect the iBGP learned routes to other iBGP speakers.

Ethernet VPN (EVPN) Autodiscovery supports BGP route reflectors. A BGP route reflector can be used to reflect BGP EVPN prefixes without EVPN being explicitly configured on the route reflector. The route reflector does not participate in autodiscovery; that is, no pseudowires are set up between the route reflector and the provider edge (PE) devices. A route reflector reflects EVPN prefixes to other PE devices so that these PE devices do not need to have a full mesh of BGP sessions. The network administrator configures only the BGP EVPN address family on a route reflector.

BGP uses the Layer 2 VPN (L2VPN) routing information base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

EVPN Address Family

BGP supports Layer 2 VPN (L2VPN) EVPN address family under router configuration mode to carry L2VPN EVPN autodiscovery and signaling Network Layer Reachability Information (NLRI) to Boarder Gateway Protocol (BGP) neighbors. This address family is allowed on both internal BGP (iBGP) and external BGP (eBGP) neighbors under default virtual routing and forwarding (VRF) for both IPv4 and IPv6 neighbors. The EVPN SAFI is not supported under VRF and VRF neighbors.

How to Configure BGP PBB EVPN Route Reflector Support

Configuring BGP PBB EVPN Route Reflector

Perform this task on the Boarder Gateway Protocol (BGP) route reflector to configure the device as a BGP route reflector and configure the specified neighbor as its client and to display the information from the BGP routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family l2vpn [vpls | evpn]**
5. **neighbor** {*ip-address* | *peer-group-name*} **activate**
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
7. **end**
8. **show bgp l2vpn evpn all**
9. **debug bgp l2vpn evpn updates**
10. **clear bgp l2vpn evpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode.
Step 4	**address-family l2vpn [vpls | evpn]** **Example:** `Device(config-router)# address-family l2vpn evpn`	Specifies the L2VPN address family and enters address family configuration mode. The optional **evpn** keyword specifies that EVPN endpoint provisioning information is to be distributed to BGP peers.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 activate`	Enables PBB EVPN with the specified BGP neighbor.
Step 6	**neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client** **Example:** `Device(config-router-af)# neighbor 10.0.0.2 route-reflector-client`	Configures the local device as the BGP route reflector and the specified neighbor as one of its clients.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show bgp l2vpn evpn all** **Example:** `Device# show bgp l2vpn evpn all`	(Optional) Displays the complete L2VPN EVPN database.
Step 9	**debug bgp l2vpn evpn updates** **Example:** `Device# debug bgp l2vpn evpn updates events`	(Optional) Specifies debugging messages for BGP update.
Step 10	**clear bgp l2vpn evpn** **Example:** `Device# clear bgp l2vpn evpn *`	(Optional) Specifies that all current BGP sessions will be reset.

Configuration Examples for BGP PBB EVPN Route Reflector Support

Example: Configuring BGP PBB EVPN Route Reflector

In the following example, the local device is a route reflector. It passes learned iBGP routes to the neighbor at 10.0.0.2:

```
Device# configure terminal
Device(config)# router bgp 1
Device(config-router)# address-family l2vpn evpn
Device(config-router-af)# neighbor 10.0.0.2 activate
Device(config-router-af)# neighbor 10.0.0.2 route-reflector-client
Device(config-router-af)# exit address-family
```

In the following example, the **show bgp l2vpn evpn all route-type 1** command displays the Ethernet autodiscovery route information:

```
show bgp l2vpn evpn all route-type 1

BGP routing table entry for
[1][100.100.100.100:11111][AAAABBBBCCCCDDDDEEEE][23456789][101234]/25, version 2
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
     1          2          3
  Refresh Epoch 1
  Local, (Received from a RR-client)
    19.0.101.1 from 19.0.101.1 (19.0.101.1)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:100:101 EVPN LABEL:0x1:Label-101234
      rx pathid: 0, tx pathid: 0x0
```

Additional References for BGP PBB EVPN Route Reflector Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 4456	*BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*
RFC 4684	*Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP PBB EVPN Route Reflector Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 147: Feature Information for BGP PBB EVPN Route Reflector Support

Feature Name	Releases	Feature Information
BGP PBB EVPN Route Reflector Support		The BGP PBB EVPN Route Reflector Support feature provides Boarder Gateway Protocol (BGP) route reflector functionality for Ethernet VPN (EVPN) and provider backbone bridging (PBB) EVPN of Layer 2 VPN address family. EVPN enables customer MAC addresses as routable addresses and distributes them in BGP to avoid any data plane MAC address learning over the Multiprotocol Label Switching (MPLS) core network. The route reflector is enhanced to store the received EVPN updates without configuring EVPN explicitly on the route reflector and then advertises these updates to other provider edge (PE) devices so that the PEs do not need to have a full mesh of BGP sessions. The following command was modified: **address-family l2vpn**.

CHAPTER 118

Overview BGP Monitoring Protocol

The BGP Monitoring Protocol (BMP) feature supports the following functionalities to monitor Border Gateway Protocol (BGP) neighbors, also called BMP clients:

- Configure devices to function as BMP servers, and on the servers, set up parameters that are required for monitoring the BGP neighbors.
- Establish connectivity of the BMP servers with BGP neighbors for monitoring.
- Generate a statistics report based on monitoring the BGP neighbors.
- Perform appropriate error handling on the BGP neighbors.
- Perform graceful scale up and degradation to the point of closing connectivity between the BMP servers and BGP neighbors.

Prerequisites for BGP Monitoring Protocol

Before you configure BGP Monitoring Protocol (BMP) servers, you must configure BGP neighbors that function as BMP clients, and establish a session with its peers using either IPv4 or IPv6, or VPNv4 or VPNv6 address family identifiers.

Information About BGP Monitoring Protocol

The BGP Monitoring Protocol (BMP) feature enables monitoring of BGP neighbors (called BMP clients). You can configure a device to function as a BMP server that monitors BMP clients, which in turn, have several active peer sessions configured. You can also configure a BMP client to connect to one or more BMP servers. The BMP feature enables the configuration of multiple BMP servers (configured as primary servers) to function actively and independent of each other simultaneously to monitor BMP clients.

Each BMP server is specified by a number, and you can use CLI to configure parameters such as IP address, port number, and so on. Upon activation, a BMP server attempts to connect to BMP clients by sending an initiation message. The CLI enables multiple (independent and asynchronous) BMP server connections.

BGP neighbors are configured to send data to specific BMP servers for monitoring purposes. These clients are configured in a queue. When a request for a connection arrives from BMP clients to BMP servers, a connection is established based on the order in which the requests arrived. After a BMP server connects with the first BMP neighbor, it sends out refresh requests to monitor the BMP clients and starts monitoring those BMP clients with whom the connection is already established.

Session connection requests from the other BMP clients to the BMP servers is initiated after an initial delay that you can configure using the **initial-delay** command. If a connection is established but fails later, the connection request is retried after a delay, which you can configure using the **failure-retry-delay** command. If there is repeated failure establishing a connection, the connection retries are delayed based on the delay configured using the **flapping-delay** command. Configuring the delay for such requests becomes significant because the route refresh requests that are sent to all the connected BMP clients causes considerable network traffic and load on the device.

To avoid excessive load on the device, the BMP servers send route-refresh requests to individual BMP clients at a time - in the order in which connections are established in the queue. After a BMP client that is already connected is in the *reporting* state, it sends a “peer-up” message to the BMP server. After the client receives a route-refresh request, route monitoring begins for that neighbor. After the route-refresh request ends, the next neighbor in the queue is processed. This cycle continues until all BGP neighbors in the reporting state are reported and all routes sent by these “reporting” BGP neighbors are monitored. If a neighbor is established after BMP monitoring begins, it does not require a route-refresh request. All the received routes from that client is sent to the BMP servers.

It is advantageous to batch up refresh requests from BMP clients if several BMP servers are activated in quick succession. Use the **bmp initial-refresh delay** command to configure a delay in triggering the refresh mechanism when the first BMP server comes up. If other BMP servers come online within this timeframe, only one set of refresh requests is sent to the BMP clients. You can also configure the **bmp initial-refresh skip** command to skip all the refresh requests from the BMP servers and just monitor all incoming messages from the peers.

In a client-server configuration, we recommend that the resource load of the devices be kept minimal and adding excessive network traffic be avoided. In the BMP configuration, you can configure various delay timers on the BMP server to avoid flapping during the connection between a server and a client. To avoid excessive message throughput or high usage of system resources, you can configure the maximum buffer limit for the BMP session.

From Cisco IOS XE Bengaluru 17.6.1, system time is used as a timestamp, and it is included in all the BMP messages with a per-peer header. When a BMP server or peer flaps, check the timestamp in the BMP message and use this information to troubleshoot the issue. The following BMP messages contain a timestamp:

- Route Monitoring
- Peer Up Notification
- Peer Down Notification
- State Reporting

These BMP messages contain timestamps by default. To verify if timestamps are included, use the **show ip bgp bmp server summary** command and verify that the output includes the following message - `BGP Message Timestamp will be sent to BMP Servers`. For more information, see Examples for Configuring, Verifying, and Monitoring BGP Monitoring Protocol section.

How to Configure BGP Monitoring Protocol

The following sections provide information about the various tasks that comprise the BMP configuration process.

Configuring a BGP Monitoring Protocol Session

Perform this task to configure the BMP session parameters for the BMP servers to establish connectivity with BMP clients.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bmp** {**buffer-size** *buffer-bytes* | **initial-refresh** {**delay** *refresh-delay* | **skip**} | **server** *server-number-n*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**bmp** {**buffer-size** *buffer-bytes* | **initial-refresh** {**delay** *refresh-delay* | **skip**} | **server** *server-number-n* **Example:** `Device(config-router)# bmp initial-refresh delay 30`	Configures BMP parameters for BGP neighbors, and enters BMP server configuration mode to configure BMP servers.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Configuring BGP Monitoring Protocol on BGP Neighbors

Perform this task to activate BMP on BGP neighbors (also called BMP clients) so that client activity is monitored by the BMP server configured on the neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ipv4-addr* | *neighbor-tag* | *ipv6-addr*} **bmp-activate** {**all** | **server** *server-number-1* [**server** *server-number-2* . . . [**server** *server-number-n*]]}
   - Repeat Step 1 to Step 4 to configure the other BMP clients in the session.
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**neighbor** {*ipv4-addr* | *neighbor-tag* | *ipv6-addr*} **bmp-activate** {**all** | **server** *server-number-1* [**server** *server-number-2* . . . [**server** *server-number-n*]]} • Repeat Step 1 to Step 4 to configure the other BMP clients in the session. **Example:** `Device(config-router)# neighbor 30.1.1.1 bmp-activate server 1 server 2`	Activates BMP monitoring on a BGP neighbor.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Configuring a BGP Monitoring Protocol Server

Perform this task to configure a BMP server and its parameters in BMP server configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bmp** {**buffer-size** *buffer-bytes* | **initial-refresh** {**delay** *refresh-delay* | **skip**} | **server** *server-number-n*
5. **activate**
6. **address** {*ipv4-addr* | *ipv6-addr*} **port-number** *port-number*
7. **description LINE** *server-description*
8. **failure-retry-delay** *failure-retry-delay*
9. **flapping-delay** *flap-delay*
10. **initial-delay** *initial-delay-time*
11. **set ip dscp** *dscp-value*
12. **stats-reporting-period** *report-period*
13. **update-source** *interface-type interface-number*
14. **exit-bmp-server-mode**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**bmp** {**buffer-size** *buffer-bytes*	**initial-refresh** {**delay** *refresh-delay*	**skip**}	**server** *server-number-n* **Example:** `Device(config-router)# bmp server 1`	Enters BMP server configuration mode to configure the BMP server.

	Command or Action	Purpose
Step 5	**activate** **Example:** `Device(config-router-bmpsrvr)# activate`	Initiates a connection between the BMP server and BGP neighbors.
Step 6	**address** {*ipv4-addr* | *ipv6-addr*} **port-number** *port-number* **Example:** `Device(config-router-bmpsrvr)# address 10.1.1.1 port-number 8000`	Configures the IP address and port number to the BMP server.
Step 7	**description LINE** *server-description* **Example:** `Device(config-router-bmpsrvr)# description LINE SERVER1`	Configures a textual description of the BMP server.
Step 8	**failure-retry-delay** *failure-retry-delay* **Example:** `Device(config-router-bmpsrvr)# failure-retry-delay 40`	Configures delay in the retry requests during failures when sending BMP server updates.
Step 9	**flapping-delay** *flap-delay* **Example:** `Device(config-router-bmpsrvr)# flapping-delay 120`	Configures delays in flapping when sending BMP server updates.
Step 10	**initial-delay** *initial-delay-time* **Example:** `Device(config-router-bmpsrvr)# initial-delay 20`	Configures delays in sending initial requests for updates from the BMP server.
Step 11	**set ip dscp** *dscp-value* **Example:** `Device(config-router-bmpsrvr)# set ip dscp 5`	Configures the IP Differentiated Services Code Point (DSCP) values for the BMP server.
Step 12	**stats-reporting-period** *report-period* **Example:** `Device(config-router-bmpsrvr)# stats-reporting-period 30`	Configures the time interval in which the BMP server receives the statistics report from BGP neighbors.
Step 13	**update-source** *interface-type interface-number* **Example:**	Configures the interface source for routing updates on the BMP servers.

	Command or Action	Purpose
	`Device(config-router-bmpsrvr)# update-source ethernet 0/0`	
Step 14	**exit-bmp-server-mode** **Example:** `Device(config-router-bmpsrvr)# exit-bmp-server-mode`	Exits from BMP server configuration mode and returns to router configuration mode. Repeat Step 1 to Step 14 to configure the other BMP servers in the session.
Step 15	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Verifying BGP Monitoring Protocol

Perform the following steps to verify the configuration for the BMP servers and BMP clients:

SUMMARY STEPS

1. **enable**
2. **show ip bgp bmp**
3. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**show ip bgp bmp** **Example:** `Device# show ip bgp bmp neighbors`	Displays information about BMP servers and neighbors.
Step 3	**show running-config** **Example:** `Device# show running-config | section bmp`	Displays information about BMP servers and neighbors.

Monitoring BGP Monitoring Protocol

Perform the following steps to enable debugging and monitor the BMP servers.

SUMMARY STEPS

1. **enable**
2. **debug ip bgp bmp**
3. **show debugging**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**debug ip bgp bmp** **Example:** `Device# debug ip bgp bmp server`	Enables debugging of the BMP attributes.
Step 3	**show debugging** **Example:** `Device# show debugging`	Displays information about the types of debugging that are enabled on a device.

Configuration Examples for BGP Monitoring Protocol

The following sections contain examples relating to configuring, verifying, and monitoring BMP.

Examples: Configuring BGP Monitoring Protocol

Note Two levels of configuration are required for the BMP to function as designed. You must enable BMP monitoring on each BGP neighbor (also called BMP client) to which several peers are connected in a network, and establish connectivity between the BMP servers and clients. Then, configure each BMP server in BMP server configuration mode for a specific server with the parameters required for monitoring the associated BMP clients.

The following example shows how to activate BMP on a neighbor with IP address 192.168.1.1, which is monitored by BMP servers (in this case, server 1 and 2):

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# neighbor 192.168.1.1 bmp-activate server 1 server 2
Device(config-router)# end
```

The following example shows how to configure an initial refresh delay of 30 seconds for BGP neighbors on which BMP is activated using the **neighbor bmp-activate** command:

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# bmp initial-refresh delay 30
Device(config-router)# bmp buffer-size 2048
Device(config-router)# end
```

The following example shows how to enter BMP server configuration mode and initiate connection between a specific BMP server with the BGP BMP neighbors. In this example, connection to clients is initiated from BMP servers 1 and 2 along with configuration of the monitoring parameters.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# bmp server 1
Device(config-router-bmpsrvr)# activate
Device(config-router-bmpsrvr)# address 10.1.1.1 port-number 8000
Device(config-router-bmpsrvr)# description LINE SERVER1
Device(config-router-bmpsrvr)# failure-retry-delay 40
Device(config-router-bmpsrvr)# flapping-delay 120
Device(config-router-bmpsrvr)# initial-delay 20
Device(config-router-bmpsrvr)# set ip dscp 5
Device(config-router-bmpsrvr)# stats-reporting-period 30
Device(config-router-bmpsrvr)# update-source ethernet 0/0
Device(config-router-bmpsrvr)# exit-bmp-server-mode
Device(config-router)# bmp server 2
Device(config-router-bmpsrvr)# activate
Device(config-router-bmpsrvr)# address 20.1.1.1 port-number 9000
Device(config-router-bmpsrvr)# description LINE SERVER2
Device(config-router-bmpsrvr)# failure-retry-delay 40
Device(config-router-bmpsrvr)# flapping-delay 120
Device(config-router-bmpsrvr)# initial-delay 20
Device(config-router-bmpsrvr)# set ip dscp 7
Device(config-router-bmpsrvr)# stats-reporting-period 30
Device(config-router-bmpsrvr)# update-source ethernet 2/0
Device(config-router-bmpsrvr)# exit-bmp-server-mode
Device(config-router)# end
```

Examples: Verifying BGP Monitoring Protocol

The following is a sample output from the **show ip bgp bmp server** command for server number 1. The attributes displayed are configured in BMP server configuration mode.

```
Device# show ip bgp bmp server 1

Print detailed info for 1 server number 1.

bmp server 1
address: 192.168.1.1    port 8000
description SERVER1
```

```
up time 00:06:22
session-startup route-refresh
initial-delay 20
failure-retry-delay 40
flapping-delay 120
activated
```

The following is a sample output from the **show ip bgp bmp server** command for server number 2. The attributes displayed are configured in BMP server configuration mode.

```
Device# show ip bgp bmp server 2

Print detailed info for 1 server number 2.

bmp server 2
address: 20.1.1.1    port 9000
description SERVER2
up time 00:06:23
session-startup route-refresh
initial-delay 20
failure-retry-delay 40
flapping-delay 120
activated
```

The following is a sample output from the **show ip bgp bmp server summary** command after deactivating the BMP servers 1 and 2 connections.

```
Device# show ip bgp bmp server summary

Number of BMP servers configured: 2
Number of BMP neighbors configured: 10
Number of neighbors on TransitionQ: 0, MonitoringQ: 0, ConfigQ: 0
Number of BMP servers on StatsQ: 0
BMP Refresh not in progress, refresh not scheduled
Initial Refresh Delay configured, refresh value 30s
BMP buffer size configured, buffer size 2048 MB, buffer size bytes used 0 MB
BGP Message Timestamp will be sent to BMP Servers

ID Host/Net          Port  TCB               Status  Uptime    MsgSent   LastStat
1  10.1.1.1          8000  0x0               Down              0
2  20.1.1.1          9000  0x0               Down              0
```

The following is a sample output from the **show ip bgp bmp neighbors** command, which shows the status of the BGP BMP neighbors after reactivating the BMP servers 1 and 2 connections.

```
Device# show ip bgp bmp server neighbors

Number of BMP neighbors configured: 10
BMP Refresh not in progress, refresh not scheduled
Initial Refresh Delay configured, refresh value 30s
BMP buffer size configured, buffer size 2048 MB, buffer size bytes used 0 MB

Neighbor           PriQ      MsgQ      CfgSvr#             ActSvr#             RM Sent
30.1.1.1           0         0         1  2                1  2                16
2001:DB8::2001     0         0         1  2                1  2                15
40.1.1.1           0         0         1  2                1  2                26
2001:DB8::2002     0         0         1  2                1  2                15
50.1.1.1           0         0         1  2                1  2                16
60.1.1.1           0         0         1  2                1  2                26
2001:DB8::2002     0         0         1                   1                   9
70.1.1.1           0         0         2                   2                   12
```

```
Neighbor           PriQ      MsgQ      CfgSvr#             ActSvr#             RM Sent
80.1.1.1           0         0         1                   1                   10
2001:DB8::2002     0         0         1  2                1  2                16
```

The following is a sample output from the **show ip bgp bmp server summary** command for BMP server number 1 and 2. The statistics reporting interval on BMP servers 1 and 2 has been set to 30 seconds. Therefore, each server receives statistics messages from its connected BGP BMP neighbor in 30-second cycles.

```
Device# show ip bgp bmp server summary

Number of BMP servers configured: 2
Number of BMP neighbors configured: 10
Number of neighbors on TransitionQ: 0, MonitoringQ: 0, ConfigQ: 0
Number of BMP servers on StatsQ: 0
BMP Refresh not in progress, refresh not scheduled
Initial Refresh Delay configured, refresh value 30s
BMP buffer size configured, buffer size 2048 MB, buffer size bytes used 0 MB
BGP Message Timestamp will be sent to BMP Servers

ID Host/Net          Port  TCB                Status  Uptime    MsgSent   LastStat
1  10.1.1.1          8000  0x2A98B07138       Up      00:38:49  162       00:00:09
2  20.1.1.1          9000  0x2A98E17C88       Up      00:38:49  46        00:00:04
```

```
Device# show ip bgp bmp server summary

Number of BMP servers configured: 2
Number of BMP neighbors configured: 10
Number of neighbors on TransitionQ: 0, MonitoringQ: 0, ConfigQ: 0
Number of BMP servers on StatsQ: 0
BMP Refresh not in progress, refresh not scheduled
Initial Refresh Delay configured, refresh value 30s
BMP buffer size configured, buffer size 2048 MB, buffer size bytes used 0 MB
BGP Message Timestamp will be sent to BMP Servers

ID Host/Net          Port  TCB                Status  Uptime    MsgSent   LastStat
1  10.1.1.1          8000  0x2A98B07138       Up      00:40:19  189       00:00:07
2  20.1.1.1          9000  0x2A98E17C88       Up      00:40:19  55        00:00:02
```

Note If you configure several BGP BMP neighbors to be monitored by the BMP servers, for example 10, then 10 statistics messages are received by both the servers during each configured cycle.

The following is a sample output from the **show running-config** command, which shows the running configuration on a device:

```
Device# show running-config | section bmp

bmp server 1
address 10.1.1.1 port-number 8000
description SERVER1
initial-delay 20
failure-retry-delay 40
flapping-delay 120
update-source Ethernet0/0
set ip dscp 3
activate
exit-bmp-server-mode
```

```
bmp server 2
address 20.1.1.1 port-number 9000
description SERVER2
initial-delay 20
failure-retry-delay 40
flapping-delay 120
update-source Ethernet2/0
set ip dscp 5
activate
exit-bmp-server-mode
bmp initial-refresh delay 30
bmp-activate all
```

Examples: Monitoring BGP Monitoring Protocol

The following examples show how to enable debugging of the various BMP attributes:

```
Device# debug ip bgp bmp event

BGP BMP events debugging is on

Device# debug ip bgp bmp neighbor

BGP BMP neighbor debugging is on

Device# debug ip bgp bmp server

BGP BMP server debugging is on
```

The following is a sample output from the **show debugging** command after you enable BGP BMP server debugging:

```
Device# show debugging

IP routing:
BGP BMP server debugging is on

Device#

*Apr  8 21:04:13.164: BGPBMP: BMP server connection attempt timer expired for server 1 -
10.1.1.1/8000
*Apr  8 21:04:13.165: BGPBMP: BMP server 1 active open process success - 10.1.1.1/8000
*Apr  8 21:04:13.165: BGPBMP: TCP KA interval is set to 15

Device#

*Apr  8 21:04:15.171: BGPBMP: Register read/write notification callbacks with BMP server 1
 TCB - 10.1.1.1/8000
*Apr  8 21:04:15.171: BGPBMP: Initiation msg sent to BMP server 1 - 10.1.1.1/8000
*Apr  8 21:04:15.171: BGPBMP: BMP server 1 connection - 10.1.1.1/8000 up, invoke refresh
event

Device#

*Apr  8 21:04:16.249: BGPBMP: BMP server connection attempt timer expired for server 2 -
20.1.1.1/9000
*Apr  8 21:04:16.249: BGPBMP: BMP server 2 active open process success - 20.1.1.1/9000
*Apr  8 21:04:16.249: BGPBMP: TCP KA interval is set to 15
*Apr  8 21:04:16.250: BGPBMP: Register read/write notification callbacks with BMP server 2
 TCB - 20.1.1.1/9000
*Apr  8 21:04:16.250: BGPBMP: Initiation msg sent to BMP server 2 - 20.1.1.1/9000
```

```
*Apr  8 21:04:16.250: BGPBMP: BMP server 2 connection - 20.1.1.1/9000 up, invoke refresh
event
```

Additional References for BGP Monitoring Protocol

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP Monitoring Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 148: Feature Information for BGP Monitoring Protocol

Feature Name	Releases	Feature Description
BGP Monitoring Protocol		

Feature Name	Releases	Feature Description
		The BGP Monitoring Protocol feature supports the following functionality to enable monitoring of the Border Gateway Protocol (BGP) neighbors, that become BMP clients: • Configure devices to function as BMP servers, and set up parameters that are required for monitoring the BGP neighbors, on the servers. • Establish connectivity of the BMP servers with BGP neighbors for monitoring. • Generate statistics report based on the task of monitoring the BGP neighbors. • Perform appropriate error handling on the BGP neighbors. • Perform graceful scale up and degradation to the point of closing connectivity between the BMP servers and BGP neighbors. The following commands were introduced or modified: **bmp** **debug ip bgp bmp** **neighbor bmp-activate** **show ip bgp bmp** The following commands were introduced in BMP server configuration mode to configure specific BMP servers: **activate** **address** **default** **description**

Feature Name	Releases	Feature Description
		exit-bmp-server-mode **failure-retry-delay** **flapping-delay** **initial-delay** **set ip dscp** **stats-reporting-period** **update-source**
BMP Per-Peer Header Timestamp	Cisco IOS XE Release 17.6.1	By default, the BMP messages with a per-peer header contain timestamps. The system time is used as a timestamp in these messages.

CHAPTER 119

VRF Aware BGP Translate-Update

The VRF aware BGP translate-update feature enables multicast forwarding on those customer-edge (CE) devices, which have an older version of Cisco software that does not support multicast BGP (mBGP) routing.

The provider-edge (PE) devices establish a virtual routing and forwarding (VRF) session with the neighbor CE devices, and configure the translate-update feature under an IPv4/IPv6 VRF address family. The PE devices translate the updates from unicast to multicast on CE devices and put them as multicast updates in the Border Gateway Protocol (BGP) VRF routing table of the PE devices for processing.

Prerequisites for VRF Aware BGP Translate-Update

- The VRF aware translate-update feature applies only to IPv4/IPv6 virtual routing and forwarding (VRF) address-families.
- You must use peer-group for the configuration of the neighbor under IPv4/IPv6 VRF address families.
- BGP neighbors that are only capable of unicast routing, must be activated under both unicast and multicast address families.
- BGP neighbors must also be enabled under the compatible multicast address family for the VRF aware translate-update feature to function as designed.
- The provider-edge (PE) devices must have multicast VRF enabled and must have a session established with the customer-edge (CE) devices.

Restrictions for VRF Aware BGP Translate-Update

- You must not configure (nonVRF) IPv4/IPv6 address families for the VRF aware BGP translate-update feature. The IPv4/IPv6 address family must be configured for multicast routing using the Subsequent Address Family Identifier (SAFI) feature.
- The VRF aware BGP translate-update feature does not support configuration of BGP neighbor using peer-template.

Information About VRF Aware BGP Translate-Update

VRF Aware BGP Translate-Update Overview

The VRF aware BGP translate-update feature enables multicast forwarding on those customer-edge (CE) devices, which have an older version of Cisco software that does not support multicast BGP (mBGP) routing.

This feature is analogous to the Subsequent Address Family Identifier (SAFI), which provides the capability to support multicast routing in the service provider's core IPv4 network, but is limited in support to IPv4/IPv6 address families. In the case of the virtual routing and forwarding (VRF) aware BGP translate-update feature, provider-edge (PE) devices establish a VRF session with the neighbor CE devices, and have the translate-update feature configured under an IPv4/IPv6 VRF address family.

When the **neighbor translate-update** command is configured on a PE device under the (IPv4 VRF) address-family configuration mode or the (IPv6 VRF) address-family configuration mode, the PE devices translate the updates from unicast to multicast on CE devices and put them in the Border Gateway Protocol (BGP) VRF routing table of the PE devices, as multicast updates, for processing. If you also configure the optional keyword **unicast**, the updates that are not translated, are placed in the PE device's unicast queue and populates the unicast VRF BGP table. The translation from unicast to multicast routes occurs from CE devices to PE devices only, and the multicast and unicast prefixes are only advertised from the CE device to the PE device's multicast neighbors.

For example, when you configure the VRF aware BGP translate-update feature under a VRF (v1) for a neighbor CE device (CE1), a neighbor topology under the IPv4-multicast-VRF or IPv6-multicast-VRF address-family is added to CE1's session with a PE device (PE1). The multicast-VRF neighbor topology does not actively participate in these multicast sessions and only forwards announcements that arrive from CE1. Once such announcements arrive, they are translated into multicast and placed in the nonactive multicast VRF neighbor's routing table. The Cisco software ensures that the routes advertised by CE1 configured under the IPv4/IPv6 VRF address-family are available on PE1's IPv4/IPv6 multicast VRF v1 address-family BGP table. These routes, along with PE1's IPv4/IPv6 multicast VRF v1 address-family BGP table, are advertised to PE1's multicast peers if you have configured the **neighbor translate-update** command . The routes are also advertised to PE1's unicast peers if you have also configured the optional keyword **unicast**.

The **unicast** keyword is optional, yet significant, as it enables the PE devices to place unicast advertisements from the CE devices in the unicast BGP table of the PE devices. Therefore, route advertisements from CE devices populates both unicast and multicast BGP tables, else CE device's routes only populate the PE device's multicast BGP table.

Note You must also enable address-family under the compatible multicast address-family for VRF aware BGP translate-update feature to function as designed.

How To Configure VRF Aware BGP Translate-Update

Configuring VRF Aware BGP Translate-Update

Perform this task to configure VRF aware BGP translate-update feature:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**mdt** | **tunnel** | {**multicast** | **unicast**} [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **neighbor** *peer-group-name* **peer-group**
6. **neighbor** {*ipv4-addr* | *ipv6-addr* | *peer-group-name*} **remote-as** *autonomous-system-number*
7. **neighbor** {*ipv4-addr* | *ipv6-addr*} **peer-group** *peer-group-name*
8. **neighbor** {*ipv4-addr* | *ipv6-addr* | *peer-group-name*} **activate**
9. **neighbor** {*ipv4-address* | *ipv6-address*} **translate-update multicast** [**unicast**]
10. **end**
11. **show bgp vpnv4 multicast** {**all** | **vrf** *vrf-name* | **rd** *route-distinguisher*}
12. **show ip route multicast vrf** *vrf-name*
13. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode and creates a BGP routing process.

	Command or Action	Purpose
Step 4	**address-family ipv4** [**mdt** | **tunnel** | {**multicast** | **unicast**} [**vrf** *vrf-name*] | **vrf** *vrf-name*] **Example:** `Device(config)# address-family ipv4 vrf v1`	Enters address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes.
Step 5	**neighbor** *peer-group-name* **peer-group** **Example:** `Device(config-af)# neighbor n2 peer-group`	Creates a BGP or multiprotocol BGP peer group.
Step 6	**neighbor** {*ipv4-addr* | *ipv6-addr* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-af)# neighbor n2 remote-as 4`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 7	**neighbor** {*ipv4-addr* | *ipv6-addr*} **peer-group** *peer-group-name* **Example:** `Device(config-af)# neighbor 10.1.1.1 peer-group n2`	Configures a BGP neighbor to be a member of a peer group.
Step 8	**neighbor** {*ipv4-addr* | *ipv6-addr* | *peer-group-name*} **activate** **Example:** `Device(config-af)# neighbor 10.1.1.1 activate`	Enables exchange of information with a BGP neighbor.
Step 9	**neighbor** {*ipv4-address* | *ipv6-address*} **translate-update multicast** [**unicast**] **Example:** `Device(config-af)# neighbor 10.1.1.1 translate-update multicast unicast`	Enables multicast routing on devices, which are not capable of multicast BGP (mBGP) routing.
Step 10	**end** **Example:** `Device(config-af)# end`	Returns to privileged EXEC mode.
Step 11	**show bgp vpnv4 multicast** {**all** | **vrf** *vrf-name* | **rd** *route-distinguisher*} **Example:** `Device# show bgp vpnv4 mul vrf v1 summary`	Displays Virtual Private Network Version 4 (VPNv4) multicast entries in a BGP table.

	Command or Action	Purpose
Step 12	**show ip route multicast vrf** *vrf-name* **Example:** `Device# show ip route multicast vrf v1`	Displays the IP routing table associated with a specific multicast VPN routing and forwarding (VRF) instance.
Step 13	**show running-config** **Example:** `Device# show running-config`	Displays the running configuration on the device.

Removing the VRF Aware BGP Translate-Update Configuration

Perform this task to disable the VRF aware BGP translate-update feature:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**mdt** | **tunnel** | {**multicast** | **unicast**} [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **no neighbor** {*ipv4-address* | *ipv6-address*} **translate-update multicast** [**unicast**]
6. **end**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 65000`	Enters router configuration mode and creates a BGP routing process.
Step 4	**address-family ipv4** [**mdt**	**tunnel**	{**multicast**	**unicast**} [**vrf** *vrf-name*]	**vrf** *vrf-name*] **Example:**	Enters address family configuration mode to configure a routing session using standard IP Version 4 (IPv4) address prefixes.

	Command or Action	Purpose
	`Device(config)# address-family ipv4 vrf v1`	
Step 5	**no neighbor** {*ipv4-address* | *ipv6-address*} **translate-update multicast** [**unicast**] **Example:** `Device(config-af)# no neighbor 10.1.1.1 translate-update multicast unicast`	Disables multicast routing on devices, which are not capable of multicast BGP (mBGP) routing.
Step 6	**end** **Example:** `Device(config-af)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Displays the running configuration on the device.

Configuration Examples for VRF Aware BGP Translate-Update

Example: Configuring VRF aware BGP Translate-Update

The following example shows how to configure the translate-update feature for an IPv4 VRF address-family named v1 and BGP neighbor n2 peer-group for VRF configuration:

Note Peer-template configuration for BGP neighbor is not supported for this feature due to conflicts with the earlier versions of Cisco software.

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv4 vrf v1
Device(config-router-af)# neighbor n2 peer-group
Device(config-router-af)# neighbor n2 remote-as 4
Device(config-router-af)# neighbor 10.1.1.1 peer-group n2
Device(config-router-af)# neighbor 10.1.1.1 activate
Device(config-router-af)# neighbor 10.1.1.1 translate-update multicast unicast
Device(config-router-af)# end
```

The following is sample output from the **show bgp vpnv4 multicast vrf** command. As the VRF aware BGP translate-update feature is configured, the state of the neighbor displays “NoNeg”:

```
Device# show bgp vpnv4 multicast vrf v1 summary
```

```
BGP router identifier 10.1.3.1, local AS number 65000
BGP table version is 8, main routing table version 8
7 network entries using 1792 bytes of memory
8 path entries using 960 bytes of memory
5/3 BGP path/bestpath attribute entries using 1280 bytes of memory
3 BGP AS-PATH entries using 88 bytes of memory
2 BGP extended community entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4168 total bytes of memory
BGP activity 23/2 prefixes, 33/9 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.1.1        4            4       5      10        1    0    0 00:01:10 (NoNeg)

10.1.3.2        4            2      12      10        8    0    0 00:01:33
```

The following is sample output from the **show ip route multicast vrf** command:

Note The routes configured using the translate-update feature does not have the "+" symbol against the prefixes in the Routing Information Base (RIB) table. Appearance of the symbol in the first entry indicates that the unicast route has leaked into the multicast table. However, the second entry is a translate-update route, which appears to be a multicast route.

```
Device# show ip route multicast vrf v1

B   +    10.1.1.0/24 [20/0] via 10.1.1.1 (v1), 00:00:08

B        10.1.1.0/24 [20/0] via 10.1.1.1 (v1), 00:00:42
```

The following is sample output from the **show running-config** command:

Note The provider-edge (PE) device must activate its BGP neighbor under the multicast address-family even though the neighbor is not capable of multicast routing. If the unicast address-family identifier has the route-map configured and multicast address-family identifier has no route-map configured, the unicast route-map controls the route under the unicast table but not the route under multicast table.

```
Device# show running-config

address-family ipv4 vrf v1
redistribute connected
redistribute static
neighbor 10.1.1.1 remote-as 4
neighbor 10.1.1.1 activate
neighbor 10.1.1.1 translate-update multicast unicast
neighbor 10.1.1.1 remote-as 4
neighbor 10.1.1.1 activate
exit-address-family
!
address-family ipv4 multicast vrf v1
redistribute connected
redistribute static
```

```
neighbor 10.1.1.1 remote-as 4
neighbor 10.1.1.1 activate
neighbor 10.1.1.1 soft-reconfiguration inbound
neighbor 10.1.1.1 route-map x in
exit-address-family
```

Note The “neighbor 10.1.1.1 soft-reconfiguration inbound” and the “ neighbor 10.1.1.1 route-map x in” field in the output indicate that only the routes in the BGP multicast table are affected.

The following is sample output from the **show running-config** command when you configure a neighbor under different address-families:

Note Configuring the BGP neighbor under different address-families manipulates the unicast routes and multicast routes advertised to the neighbor.

Configuration for IPv4/IPv6 unicast address-family:

```
Device# show running-config

address-family ipv4
neighbor 20.2.2.1 activate
neighbor 20.2.2.1 translate-update multicast unicast
exit-address-family
!
address-family ipv4 multicast
neighbor 20.2.2.1 activate
exit-address-family
!
```

Configuration for IPv4/IPv6 VRF unicast address-family:

```
Device# show running-config

address-family ipv4 vrf v1
neighbor 20.2.2.1 remote-as 4
neighbor 20.2.2.1 activate
neighbor 20.2.2.1 translate-update multicast unicast
exit-address-family
!
address-family ipv4 multicast vrf v1
neighbor 20.2.2.1 remote-as 4
neighbor 20.2.2.1 activate
exit-address-family
!
```

The following is sample configuration of the translate-update feature from a device with the old version of Cisco Software. The neighbor, in this case, is configured for IPv4/IPv6 unicast address-family, without running the **address-family** command:

Configuration in the old format, without an address-family configured:

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
```

```
Device(config-router)# neighbor 20.2.2.1 remote-as 4
Device(config-router)# neighbor 20.2.2.1 translate-update nlri ipv4 multicast unicast
Device(config-router-af)# end
```

Configuration in the new format, without an address-family configured:

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# neighbor 20.2.2.1 remote-as 4
Device(config-router)# neighbor 20.2.2.1 translate-update nlri multicast unicast
Device(config-router-af)# end
```

Example: Removing VRF aware BGP Translate-Update Configuration

The following example shows how to disable the VRF aware BGP translate-update feature for an IPv4 VRF address-family named v1 and BGP neighbor n2 peer-group for VRF:

Note Disabling the translate-update configuration for a neighbor deletes the pseudo multicast neighbor and flaps the session, similar to removing the neighbor from a multicast session:

```
Device> enable
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv4 vrf v1
Device(config-router-af)# no neighbor 10.1.1.1 translate-update multicast unicast
Device(config-router-af)# end
```

The following output displays the debug logs after you disable the translate-update feature on the neighbor:

```
*Nov 20 07:09:15.902: %BGP_SESSION-5-ADJCHANGE:
neighbor 2.2.2.1 IPv4 Multicast vpn vrf v1 topology base removed from session  Neighbor
deleted
*Nov 20 07:09:15.902: %BGP-5-ADJCHANGE:
neighbor 2.2.2.1 vpn vrf v1 Down Neighbor deleted
*Nov 20 07:09:15.902: %BGP_SESSION-5-ADJCHANGE:
neighbor 2.2.2.1 IPv4 Unicast vpn vrf v1 topology base removed from session  Neighbor deleted
*Nov 20 07:09:16.877: %BGP-5-ADJCHANGE:
neighbor 2.2.2.1 vpn vrf v1 Up
```

The following is sample output from the **show running-config** command:

Note The associated neighbor 10.1.1.1 is removed even from the nonvolatile generation (NVGEN) after translate-update is disabled on that neighbor.

```
Device# show running-config

address-family ipv4 vrf v1
redistribute connected
```

```
redistribute static
neighbor 10.1.1.1 remote-as 4
neighbor 10.1.1.1 activate
exit-address-family
!
address-family ipv4 multicast vrf v1
redistribute connected
redistribute static
exit-address-family
```

Additional References for VRF Aware BGP Translate-Update

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for VRF Aware BGP Translate-Update

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 149: Feature Information for VRF Aware BGP Translate-Update

Feature Name	Releases	Feature Information
VRF aware BGP Translate-Update		The VRF aware BGP translate-update feature enables multicast forwarding on those customer-edge (CE) devices, which have an older version of Cisco software that does not support multicast BGP (mBGP) routing. The following command was introduced: **neighbor translate-update**

CHAPTER 120

BGP Support for MTR

The BGP Support for MTR feature provides Border Gateway Protocol (BGP) support for multiple logical topologies over a single physical network. This module describes how to configure BGP for Multitopology Routing (MTR).

Prerequisites for BGP Support for MTR

- Be familiar with all the concepts in the “Information About BGP Support for MTR” section.
- Configure and activate a global Multitopology Routing (MTR) topology configuration.

Restrictions for BGP Support for MTR

- Redistribution within a topology is permitted. Redistribution from one topology to another is not permitted. This restriction is designed to prevent routing loops. You can use topology translation or topology import functionality to move routes from one topology to another.
- Only a single multicast topology can be configured, and only the base topology can be specified if a multicast topology is created.

Information About BGP Support for MTR

Routing Protocol Support for MTR

You must enable IP routing on the device for Multitopology Routing (MTR) to operate. MTR supports static and dynamic routing in Cisco software. You can enable dynamic routing per topology to support interdomain and intradomain routing. Route calculation and forwarding are independent for each topology. MTR support is integrated into Cisco software for the following protocols:

- Border Gateway Protocol (BGP)
- Integrated Intermediate System-to-Intermediate System (IS-IS)

You apply the per-topology configuration in router address family configuration mode of the global routing process (router configuration mode). The address family and subaddress family are specified when the device enters address family configuration mode. You specify the topology name and topology ID by entering the **topology** command in address family configuration mode.

You configure each topology with a unique topology ID under the routing protocol. The topology ID is used to identify and group Network Layer Reachability Information (NLRI) for each topology in updates for a given protocol. In OSPF, EIGRP, and IS-IS, you enter the topology ID during the first configuration of the **topology** command for a class-specific topology. In BGP, you configure the topology ID by entering the **bgp tid** command under the topology configuration.

You can configure class-specific topologies with different metrics than the base topology. Interface metrics configured on the base topology can be inherited by the class-specific topology. Inheritance occurs if no explicit inheritance metric is configured in the class-specific topology.

You configure BGP support only in router configuration mode. You configure Interior Gateway Protocol (IGP) support in router configuration mode and in interface configuration mode.

By default, interfaces are not included in nonbase topologies. For routing protocol support for EIGRP, IS-IS, and OSPF, you must explicitly configure a nonbase topology on an interface. You can override the default behavior by using the **all-interfaces** command in address family topology configuration mode. The **all-interfaces** command causes the nonbase topology to be configured on all interfaces of the device that are part of the default address space or the virtual routing and forwarding (VRF) instance in which the topology is configured.

BGP Network Scope

To implement Border Gateway Protocol (BGP) support for Multitopology Routing (MTR), the scope hierarchy is required, but the scope hierarchy is not limited to MTR use. The scope hierarchy introduces new configuration modes such as router scope configuration mode. The device enters router scope configuration mode when you configure the **scope** command in router configuration mode. When this command is entered, a collection of routing tables is created.

You configure BGP commands under the scope hierarchy for a single network (globally), or on a per-virtual routing and forwarding (VRF) basis; these configurations are referred to as scoped commands. The scope hierarchy can contain one or more address families.

MTR CLI Hierarchy Under BGP

The Border Gateway Protocol (BGP) CLI provides backward compatibility for pre-Multitopology Routing (MTR) BGP configuration and provides a hierarchical implementation of MTR. Router configuration mode is backward compatible with the pre-address family and pre-MTR configuration CLI. Global commands that affect all networks are configured in this configuration mode. For address family and topology configuration, you configure general session commands and peer templates to be used in address family configuration mode or in topology configuration mode.

After configuring any global commands, you define the scope either globally or for a specific virtual routing and forwarding (VRF) instance. The device enters address family configuration mode when you configure the **address-family** command in router scope configuration mode or in router configuration mode. Unicast is the default address family if no subaddress family identifier (SAFI) is specified. MTR supports only the IPv4 address family with a SAFI of unicast or multicast.

When the device enters address family configuration mode from router configuration mode, the software configures BGP to use pre-MTR-based CLI. This configuration mode is backward compatible with pre-existing address family configurations. Entering address family configuration mode from router scope configuration mode configures the device to use the hierarchical CLI that supports MTR. Address family configuration parameters that are not specific to a topology are entered in this address family configuration mode.

The device enters BGP topology configuration mode when you configure the **topology** command in address family configuration mode. You can configure up to 32 topologies (including the base topology) on a device. You configure the topology ID by entering the **bgp tid** command. All address family and subaddress family configuration parameters for the topology are configured here.

Note Configuring a scope for a BGP routing process removes CLI support for pre-MTR-based configuration.

The following example shows the hierarchy levels that are used when you configure BGP for MTR implementation:

```
router bgp <autonomous-system-number>
 ! Global commands

 scope {global | vrf <vrf-name>}
  ! Scoped commands

  address-family {<afi>} [<safi>]
   ! Address family specific commands

   topology {<topology-name> | base}
    ! topology specific commands
```

BGP Sessions for Class-Specific Topologies

Multitopology Routing (MTR) is configured under the Border Gateway Protocol (BGP) on a per-session basis. The base unicast and multicast topologies are carried in the global (default) session. A separate session is created for each class-specific topology that is configured under a BGP routing process. Each session is identified by its topology ID. BGP performs a best-path calculation individually for each class-specific topology. A separate Routing Information Base (RIB) and Forwarding Information Base (FIB) are maintained for each session.

Topology Translation Using BGP

Depending on the design and policy requirements for your network, you might need to install routes from a class-specific topology on one device in a class-specific topology on a neighboring device. Topology translation functionality using the Border Gateway Protocol (BGP) provides support for this operation. Topology translation is BGP neighbor-session based. You configure the **neighbor translate-topology** command by using the IP address and topology ID from the neighbor.

The topology ID identifies the class-specific topology of the neighbor. The routes in the class-specific topology of the neighbor are installed in the local class-specific Routing Information Base (RIB). BGP performs a best-path calculation on all installed routes and installs these routes into the local class-specific RIB. If a duplicate route is translated, BGP selects and installs only one instance of the route per standard BGP best-path calculation behavior.

Topology Import Using BGP

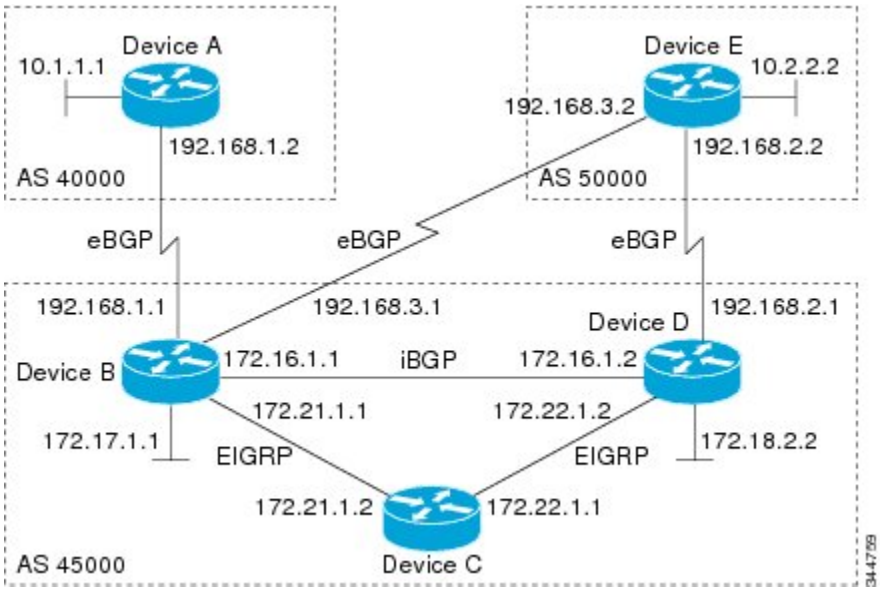
Importing topologies using the Border Gateway Protocol (BGP) is similar to topology translation. The difference is that routes are moved between class-specific topologies on the same device. You configure this function by entering the **import topology** command and specify the name of the class-specific topology or base topology. Best-path calculations are run on the imported routes before they are installed into the topology Routing Information Base (RIB). This **import topology** command also includes a **route-map** keyword to allow you to filter routes that are moved between class-specific topologies.

How to Configure BGP Support for MTR

Activating an MTR Topology by Using BGP

Perform this task to activate a Multitopology Routing (MTR) topology inside an address family by using the Border Gateway Protocol (BGP). This task is configured on Device B in the figure below and must also be configured on Device D and Device E. In this task, a scope hierarchy is configured to apply globally, and a neighbor is configured in router scope configuration mode. Under the IPv4 unicast address family, an MTR topology that applies to video traffic is activated for the specified neighbor. There is no interface configuration mode for BGP topologies.

Figure 128: BGP Network Diagram



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **scope** {**global** | **vrf** *vrf-name*}
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **transport** {**connection-mode** {**active** | **passive**} | **path-mtu-discovery** | **multi-session** | **single-session**}
7. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
8. **topology** {**base** | *topology-name*}
9. **bgp tid** *number*
10. **neighbor** *ip-address* **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **translate-topology** *number*
12. **end**
13. **clear ip bgp topology** {***** | *topology-name*} {*as-number* | **dampening** [*network-address* [*network-mask*]] | **flap-statistics** [*network-address* [*network-mask*]] | **peer-group** *peer-group-name* | **table-map** | **update-group** [*number* | *ip-address*]} [**in** [**prefix-filter**] | **out** | **soft** [**in** [**prefix-filter**] | **out**]]
14. **show ip bgp topology** {***** | *topology*} **summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**scope** {**global** | **vrf** *vrf-name*} **Example:** `Device(config-router)# scope global`	Defines the scope for the BGP routing process and enters router scope configuration mode. • BGP general session commands that apply to a single network, or a specified virtual and routing forwarding (VRF) instance, are entered in this configuration mode. • Use the **global** keyword to specify that BGP uses the global routing table. • Use the **vrf** *vrf-name* keyword and argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Device(config-router-scope)# neighbor 172.16.1.2 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local device.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **transport** {**connection-mode** {**active** | **passive**} | **path-mtu-discovery** | **multi-session** | **single-session**} **Example:** `Device(config-router-scope)# neighbor 172.16.1.2 transport multi-session`	Enables a TCP transport session option for a BGP session. • Use the **connection-mode** keyword to specify the type of connection, either active or passive. • Use the **path-mtu-discovery** keyword to enable the TCP transport path maximum transmission unit (MTU) discovery. • Use the **multi-session** keyword to specify a separate TCP transport session for each address family. • Use the **single-session** keyword to specify that all address families use a single TCP transport session.
Step 7	**address-family ipv4** [**mdt** | **multicast** | **unicast**] **Example:**	Specifies the IPv4 address family and enters router scope address family configuration mode.

	Command or Action	Purpose
	`Device(config-router-scope)# address-family ipv4`	• Use the **mdt** keyword to specify IPv4 multicast distribution tree (MDT) address prefixes. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **unicast** keyword to specify the IPv4 unicast address family. By default, the device is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • Nontopology-specific configuration parameters are configured in this configuration mode.
Step 8	**topology** {**base** | *topology-name*} **Example:** `Device(config-router-scope-af)# topology VIDEO`	Configures the topology instance in which BGP routes class-specific or base topology traffic, and enters router scope address family topology configuration mode.
Step 9	**bgp tid** *number* **Example:** `Device(config-router-scope-af-topo)# bgp tid 100`	Associates a BGP routing process with the specified topology ID. • Each topology must be configured with a unique topology ID.
Step 10	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-scope-af-topo)# neighbor 172.16.1.2 activate`	Enables the BGP neighbor to exchange prefixes for the network service access point (NSAP) address family with the local device. **Note** If you have configured a peer group as a BGP neighbor, do not use this command because peer groups are automatically activated when any peer group parameter is configured.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **translate-topology** *number* **Example:** `Device(config-router-scope-af-topo)# neighbor 172.16.1.2 translate-topology 200`	(Optional) Configures BGP to install routes from a topology on another device to a topology on the local device. • The topology ID is entered for the *number* argument to identify the topology on the device.
Step 12	**end** **Example:** `Device(config-router-scope-af-topo)# end`	(Optional) Exits router scope address family topology configuration mode and returns to privileged EXEC mode.
Step 13	**clear ip bgp topology** {***** | *topology-name*} {*as-number* | **dampening** [*network-address* [*network-mask*]] | **flap-statistics** [*network-address* [*network-mask*]] |	Resets BGP neighbor sessions under a specified topology or all topologies.

	Command or Action	Purpose
	peer-group *peer-group-name* | **table-map** | **update-group** [*number* | *ip-address*]} [**in** [**prefix-filter**] | **out** | **soft** [**in** [**prefix-filter**] | **out**]] **Example:** `Device# clear ip bgp topology VIDEO 45000`	
Step 14	**show ip bgp topology** {***** | *topology*} **summary** **Example:** `Device# show ip bgp topology VIDEO summary`	(Optional) Displays BGP information about a topology. • Most standard BGP keywords and arguments can be entered following the **topology** keyword. **Note** Only the syntax required for this task is shown. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

What to Do Next

Repeat this task for every topology that you want to enable, and repeat this configuration on all neighbor devices that are to use the topologies.

If you want to import routes from one Multitopology Routing (MTR) topology to another on the same device, see the "Importing Routes from an MTR Topology by Using BGP" section.

Importing Routes from an MTR Topology by Using BGP

Perform this task to import routes from one Multitopology Routing (MTR) topology to another on the same device, when multiple topologies are configured on the same device. In this task, a prefix list is defined to permit prefixes from the 10.2.2.0 network, and this prefix list is used with a route map to filter routes moved from the imported topology. A global scope is configured, address family IPv4 is entered, the VIDEO topology is specified, the VOICE topology is imported, and the routes are filtered using the route map named 10NET.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *number*] {**deny** | **permit**} *network/length* [**ge** *ge-length*] [**le** *le-length*]
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* [*access-list-number ...* | *access-list-name...*] | *access-list-name* [*access-list-number ...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
6. **exit**
7. **router bgp** *autonomous-system-number*
8. **scope** {**global** | **vrf** *vrf-name*}
9. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
10. **topology** {**base** | *topology-name*}
11. **import topology** {**base** | *topology-name*} [**route-map** *map-name*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *number*] {**deny** | **permit**} *network/length* [**ge** *ge-length*] [**le** *le-length*] **Example:** `Device(config)# ip prefix-list TEN permit 10.2.2.0/24`	Configures an IP prefix list. • In this example, prefix list TEN permits advertising of the 10.2.2.0/24 prefix depending on a match set by the **match ip address** command.
Step 4	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map 10NET`	Creates a route map and enters route-map configuration mode. • In this example, the route map named 10NET is created.
Step 5	**match ip address** {*access-list-number* [*access-list-number ...* | *access-list-name...*] | *access-list-name* [*access-list-number ...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Device(config-route-map)# match ip address prefix-list TEN`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match prefixes permitted by prefix list TEN.
Step 6	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 50000`	Enters router configuration mode to create or configure a Border Gateway Protocol (BGP) routing process.
Step 8	**scope** {**global** | **vrf** *vrf-name*} **Example:** `Device(config-router)# scope global`	Defines the scope to the BGP routing process and enters router scope configuration mode. • BGP general session commands that apply to a single network, or a specified virtual routing and forwarding

	Command or Action	Purpose
		(VRF) instance, are entered in this configuration mode. • Use the **global** keyword to specify that BGP uses the global routing table. • Use the **vrf** *vrf-name* keyword and argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.
Step 9	**address-family ipv4** [**mdt** | **multicast** | **unicast**] **Example:** `Device(config-router-scope)# address-family ipv4`	Enters router scope address family configuration mode to configure an address family session under BGP. • Nontopology-specific configuration parameters are configured in this configuration mode.
Step 10	**topology** {**base** | *topology-name*} **Example:** `Device(config-router-scope-af)# topology VIDEO`	Configures the topology instance in which BGP routes class-specific or base topology traffic, and enters router scope address family topology configuration mode.
Step 11	**import topology** {**base** | *topology-name*} [**route-map** *map-name*] **Example:** `Device(config-router-scope-af-topo)# import topology VOICE route-map 10NET`	(Optional) Configures BGP to move routes from one topology to another on the same device. • The **route-map** keyword can be used to filter routes that moved between topologies.
Step 12	**end** **Example:** `Device(config-router-scope-af-topo)# end`	(Optional) Exits router scope address family topology configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP Support for MTR

Example: BGP Topology Translation Configuration

The following example shows how to configure the Border Gateway Protocol (BGP) in the VIDEO topology and how to configure topology translation with the 192.168.2.2 neighbor:

```
router bgp 45000
 scope global
  neighbor 172.16.1.1 remote-as 50000
  neighbor 192.168.2.2 remote-as 55000
  neighbor 172.16.1.1 transport multi-session
  neighbor 192.168.2.2 transport multi-session
   address-family ipv4
    topology VIDEO
```

```
     bgp tid 100
     neighbor 172.16.1.1 activate
     neighbor 192.168.2.2 activate
     neighbor 192.168.2.2 translate-topology 200
     end
clear ip bgp topology VIDEO 50000
```

Example: BGP Global Scope and VRF Configuration

The following example shows how to configure a global scope for a unicast topology and also for a multicast topology. After the device exits the router scope configuration mode, a scope is configured for the virtual routing and forwarding (VRF) instance named DATA.

```
router bgp 45000
 scope global
  bgp default ipv4-unicast
  neighbor 172.16.1.2 remote-as 45000
  neighbor 192.168.3.2 remote-as 50000
  address-family ipv4 unicast
   topology VOICE
   bgp tid 100
   neighbor 172.16.1.2 activate
   exit
  address-family ipv4 multicast
   topology base
    neighbor 192.168.3.2 activate
    exit
   exit
  exit
 scope vrf DATA
  neighbor 192.168.1.2 remote-as 40000
  address-family ipv4
   neighbor 192.168.1.2 activate
   end
```

Examples: BGP Topology Verification

The following example shows summary output for the **show ip bgp topology** command. Information is displayed about Border Gateway Protocol (BGP) neighbors configured to use the Multitopology Routing (MTR) topology named VIDEO.

```
Device# show ip bgp topology VIDEO summary

BGP router identifier 192.168.3.1, local AS number 45000
BGP table version is 1, main routing table version 1
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.1.2      4 45000     289     289        1    0    0 04:48:44        0
192.168.3.2     4 50000       3       3        1    0    0 00:00:27        0
```

The following partial output displays BGP neighbor information under the VIDEO topology:

```
Device# show ip bgp topology VIDEO neighbors 172.16.1.2

BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 192.168.2.1
  BGP state = Established, up for 04:56:30
  Last read 00:00:23, last write 00:00:21, hold time is 180, keepalive interval is 60
seconds
```

```
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
  Message statistics, state Established:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                0          0
    Keepalives:           296        296
    Route Refresh:          0          0
    Total:                297        297
  Default minimum time between advertisement runs is 0 seconds
 For address family: IPv4 Unicast topology VIDEO
  Session: 172.16.1.2 session 1
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
1 update-group member
  Topology identifier: 100
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Minimum incoming TTL 0, Outgoing TTL 255
Local host: 172.16.1.1, Local port: 11113
Foreign host: 172.16.1.2, Foreign port: 179
.
.
.
```

Example: Importing Routes from an MTR Topology by Using BGP

The following example shows how to configure an access list to be used by a route map named VOICE to filter routes imported from the Multitopology Routing (MTR) topology named VOICE. Only routes with the prefix 192.168.1.0 are imported.

```
access-list 1 permit 192.168.1.0 0.0.0.255
route-map BLUE
 match ip address 1
 exit
router bgp 50000
 scope global
  neighbor 10.1.1.2 remote-as 50000
  neighbor 172.16.1.1 remote-as 60000
   address-family ipv4
    topology VIDEO
     bgp tid 100
     neighbor 10.1.1.2 activate
     neighbor 172.16.1.1 activate
     import topology VOICE route-map VOICE
     end
clear ip bgp topology VIDEO 50000
```

Additional References

Related Documents

Related Topic	Document Title
Multitopology Routing (MTR) commands	Cisco IOS Multitopology Routing Command Reference
Border Gateway Protocol (BGP) commands	Cisco IOS IP Routing: BGP Command Reference
BGP concepts and tasks	*IP Routing: BGP Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BGP Support for MTR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 150: Feature Information for BGP Support for MTR

Feature Name	Releases	Feature Information
BGP Support for MTR	12.2(33)SRB 15.0(1)S	This feature provides Border Gateway Protocol (BGP) support for multiple logical topologies over a single physical network. In Cisco IOS XE Release 2.5, support was added for the Cisco ASR 1000 Series Routers. The following commands were introduced or modified: **address-family ipv4**, **bgp tid**, **clear ip bgp topology**, **import topology**, **neighbor translate-topology**, **neighbor transport**, **scope**, **show ip bgp topology**, **topology**.

CHAPTER 121

BGP Accumulated IGP

The BGP Accumulated IGP feature is an optional nontransitive Border Gateway Protocol (BGP) path attribute. The attribute type code for the accumulated interior gateway protocol (AIGP) attribute is assigned by the Internet Assigned Numbers Authority (IANA). The value field of the AIGP attribute is defined as a set of type, length, value (TLV) elements. The AIGP TLV contains the AIGP metric.

Information About BGP Accumulated IGP

Overview of BGP Accumulated IGP

The BGP Accumulated IGP feature is required to simulate the current Open Shortest Path First (OSPF) behavior of computing the distance associated with a path. OSPF or Label Distribution Protocol (LDP) carries the prefix or label information only in the local area. Then, Border Gateway Protocol (BGP) carries the prefix or label to all the remote areas by redistributing the routes into BGP at area boundaries. The routes or labels are then advertised using label-switched paths (LSP). The next-hop for the route is changed at each Area Border Router (ABR) to a local device, which removes the need to leak OSPF routes across area boundaries. The bandwidth available on each of the core links is mapped to the OSPF cost; therefore, it is imperative that BGP carries this cost correctly between each of the provider edge (PE) devices. This functionality is achieved by using the BGP Accumulated IGP feature.

You need to enable accumulated interior gateway protocol (AIGP) processing for internal Border Gateway Protocol (iBGP) and external Border Gateway Protocol (eBGP) neighbors to carry the AIGP attribute. Neighbors configured with the AIGP attribute are put in a separate update group from other iBGP neighbors. A separate update group is required for neighbors that are enabled to send the AIGP value to cost community. BGP needs to translate the AIGP attribute to the cost community or multi-exit discriminator (MED) and attach it to the route before advertising to legacy.

When BGP installs AIGP attribute routes into the routing information base (RIB), it adds the AIGP cost with the next-hop cost. If the next-hop is a nonrecursive IGP route, BGP sets the AIGP metric to the received AIGP value and the first hop IGP metric to the next-hop. If the next-hop is a recursive route with the AIGP metric, BGP adds the received AIGP metric to the next-hop AIGP metric.

Sending and Receiving BGP Accumulated IGP

When a session receives a prefix with the accumulated interior gateway protocol (AIGP) attribute and is not configured to receive AIGP information, the session discards the AIGP attribute and processes the remainder of the update message, and then it passes the AIGP attribute to other BGP peers. The route is then installed into the routing information base (RIB) and the prefix is sent with the AIGP attribute to all the AIGP-enabled neighbors. The AIGP attribute value is not updated if the next-hop of the route is not changed by the device before advertising it to the neighbor. If the device changes the next-hop of the route, it recalculates the AIGP attribute value by adding the next-hop metric to the received AIGP attribute value.

Originating Prefixes with Accumulated IGP

Origination of routes with the accumulated interior gateway protocol (AIGP) metric is controlled by configuration. AIGP attributes are attached to redistributed routes that satisfy the following conditions:

- The protocol redistributing the route is enabled for AIGP.
- The route is an interior gateway protocol (IGP) route redistributed into Border Gateway Protocol (BGP). The value assigned to the AIGP attribute is the value of the IGP next-hop to the route or as set by a route policy.
- The route is a static route redistributed into BGP. The value assigned is the value of the next-hop to the route or as set by a route policy.
- The route is imported into BGP through a network statement. The value assigned is the value of the next-hop to the route or as set by a route policy.
- The inbound or outbound route map also creates an AIGP attribute route map using the **set aigp-metric** command.

How to Configure BGP Accumulated IGP

Configuring AIGP Metric Value

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
5. **redistribute** *protocol autonomous-system-number* **route-map** *map-tag*
6. **network** *network-id* **route-map** *map-tag*
7. **exit**
8. **route-map rtmap**
9. **set aigp-metric [igp-metric |** *value***]**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 address family and enters address family configuration mode.
Step 5	**redistribute** *protocol autonomous-system-number* **route-map** *map-tag* **Example:** `Device(config-router-af)# redistribute bgp 100 route-map rtmap`	Redistributes routes from one routing domain to another routing domain.
Step 6	**network** *network-id* **route-map** *map-tag* **Example:** `Device(config-router-af)# network 10.1.1.1 route-map rtmap`	Specifies the networks to be advertised by the Border Gateway Protocol (BGP) routing process.
Step 7	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and returns to global configuration mode.
Step 8	**route-map rtmap** **Example:** `Device(config)# route-map rtmap`	Enters route map configuration mode.

	Command or Action	Purpose
Step 9	**set aigp-metric [igp-metric** | *value***]** **Example:** `Device(config-route-map)# set aigp-metric igp-metric`	Specifies a metric value for the accumulated interior gateway protocol (AIGP) attribute. The manual metric value range is from 0 to 4294967295.
Step 10	**end** **Example:** `Device(config-route-map)# end`	Exits route map configuration mode and enters privileged EXEC mode.

Enabling Send and Receive for an AIGP Attribute

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family {ipv4 | ipv6}** [**unicast**]
5. **neighbor** *ip-address* **aigp**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family {ipv4 | ipv6}** [**unicast**] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family and enters address family configuration mode.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **aigp** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 aigp`	Enables send and receive of the AIGP attribute per neighbor.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring BGP Accumulated IGP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family {ipv4 | ipv6}** [**unicast**]
5. **neighbor** *ip-address* **aigp [send {cost-community** *community-id* **poi {igp-cost | pre-bestpath} [transitive]} | med]**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Device(config)# router bgp 40000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family {ipv4 | ipv6}** [**unicast**] **Example:** `Device(config-router)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family and enters address family configuration mode.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **aigp [send {cost-community** *community-id* **poi {igp-cost | pre-bestpath} [transitive]} | med]** **Example:** `Device(config-router-af)# neighbor 192.168.1.1 aigp send med`	Translates the AIGP attribute to MED and attaches it to the route before advertising to legacy provider edge (PE) devices.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP Accumulated IGP

Example: Configuring AIGP Metric Value

The following is a sample configuration for originating prefixes with the accumulated internal gateway protocol (AIGP) metric attribute:

```
Device# configure terminal
Device(config)# router bgp 40000
Device(config-router)# address-family ipv4 unicast
Device(config-router-af)# redistribute bgp 100 route-map rtmap
Device(config-router-af)# network 10.1.1.1 route-map rtmap
Device(config-router-af)# exit
Device(config)# route-map rtmap
Device(config-route-map)# set aigp-metric igp-metric
Device(config-route-map)# end
```

Example: Enabling Send and Receive for an AIGP Attribute

The following example shows how to enable AIGP send and receive capability in address family configuration mode:

```
Device# configure terminal
Device(config)# router bgp 40000
Device(config-router)# address-family ipv4 unicast
Device(config-router-af)# neighbor 192.168.1.1 aigp
Device(config-router-af)# exit
```

Example: Configuring BGP Accumulated IGP

In the following example, the device belongs to autonomous system 65000 and is configured to send the cost-community attribute to its neighbor at IP address 172.16.70.23:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv4 multicast
Device(config-router-af)# neighbor 172.16.70.23 aigp send cost-community 100 poi igp-cost
transitive
Device(config-router-af)# exit
```

In the following example, the device belongs to autonomous system 65000 and is configured to send the MED attribute to its neighbor at IP address 172.16.70.23:

```
Device# configure terminal
Device(config)# router bgp 65000
Device(config-router)# address-family ipv4 multicast
Device(config-router-af)# neighbor 172.16.70.23 aigp send med
Device(config-router-af)# exit
```

Additional References for BGP Accumulated IGP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP Accumulated IGP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 151: Feature Information for BGP Accumulated IGP

Feature Name	Releases	Feature Information
BGP Accumulated IGP		The BGP Accumulated IGP feature is an optional nontransitive Border Gateway Protocol (BGP) path attribute. The attribute type code for the accumulated interior gateway protocol (AIGP) attribute is assigned by the IANA. The value field of the AIGP attribute is defined as a set of type, length, value (TLV) elements. The AIGP TLV contains the AIGP metric. The following commands were introduced: **aigp**, **aigp send cost-community**, **aigp send med**, **bgp bestpath aigp ignore**, **set aigp-metric**

CHAPTER 122

BGP MVPN Source-AS Extended Community Filtering

The BGP MVPN Source-AS Extended Community Filtering feature enables the provider edge (PE) device to suppress attaching the multicast VPN (MVPN)-related extended communities to routes learned from a customer edge (CE) device or redistributed in a virtual routing and forwarding (VRF) instance for a specified neighbor.

Information About BGP MVPN Source-AS Extended Community Filtering

Overview of BGP MVPN Source-AS Extended Community Filtering

VPN routes carry special extended communities (source autonomous system [AS] extended community and virtual routing and forwarding [VRF] route import extended community) to support multicast VPN (MVPN). Legacy provider edge (PE) devices interpret the source AS extended community as old style multicast distribution tree (MDT). You can attach the extended communities when the prefix is created. After the BGP MVPN Source-AS Extended Community Filtering feature is enabled, this allows the PE device to suppress these extended communities. You can use this functionality to suppress extended communities from being sent for Subsequent Address Family Identifier (SAFI) 128 routes and instead use SAFI 129. Devices with SAFI 129 must be able to identify the source AS extended community correctly.

How to Configure BGP MVPN Source-AS Extended Community Filtering

Configuring BGP MVPN Source-AS Extended Community Filtering

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **unicast-reachability** [**source-as | vrf-route-import**] [**disable**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 45000`	Enters router configuration mode for the specified routing process.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family and enters address family configuration mode. • Use the **vrf** keyword and *vrf-name* argument to specify the name of the VRF instance to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**unicast-reachability** [**source-as	vrf-route-import**] [**disable**] **Example:**	Disables advertising extended communities for non-MVPN profiles.

	Command or Action	Purpose
	`Device(config-router-af)# unicast-reachability source-as disable`	
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Configuration Examples for BGP MVPN Source-AS Extended Community Filtering

Example: Configuring BGP MVPN Source-AS Extended Community Filtering

The following example configures BGP MVPN source-AS extended community filtering:

```
Device# configure terminal
Device(config)# router bgp 45000
Device(config)# address-family ipv4 vrf vpn1
Device(config-router-af)# unicast-reachability source-as disable
Device(config-router-af)# exit
```

The following example shows summary output for the **show ip bgp vpnv4 vrf vpn1** command.

```
Device# show ip bgp vpnv4 vrf vpn1

BGP routing table entry for 10:10:1.1.1.1/32, version 25
Paths: (2 available, best #2, table red)
Multipath: eiBGP
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local, imported path from 10:11:1.1.1.1/32 (global)
    1.1.1.2 (metric 11) (via default) from 1.1.1.5 (1.1.1.5)
      Origin incomplete, metric 11, localpref 100, valid, internal
      Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000C80200
        MVPN AS:55:0.0.0.0 MVPN VRF:1.1.1.2:2 OSPF RT:0.0.0.0:2:0
        OSPF ROUTER ID:10.10.20.2:0
      Originator: 1.1.1.2, Cluster list: 1.1.1.5
      Connector Attribute: count=1
       type 1 len 12 value 10:11:1.1.1.2
      mpls labels in/out 20/21
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  Local
    10.10.10.100 (via vrf red) from 0.0.0.0 (1.1.1.1)
      Origin incomplete, metric 11, localpref 100, weight 32768, valid, sourced, best
      Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000C80200
        MVPN VRF:1.1.1.1:1 OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.10.10.1:0
      mpls labels in/out 20/nolabel
      rx pathid: 0, tx pathid: 0x0
```

Additional References for BGP MVPN Source-AS Extended Community Filtering

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference
BGP concepts and tasks	*IP Routing: BGP Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP MVPN Source-AS Extended Community Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 152: Feature Information for BGP MVPN Source-AS Extended Community Filtering

Feature Name	Releases	Feature Information
BGP MVPN Source-AS Extended Community Filtering		The BGP MVPN Source-AS Extended Community Filtering feature enables the provider edge (PE) device to suppress attaching the multicast VPN (MVPN)-related extended communities to routes learned from a customer edge (CE) device or redistributed in a virtual routing and forwarding (VRF) instance for a specified neighbor. The following command was introduced or modified: **unicast-reachability**.

CHAPTER 123

BGP AS-Override Split-Horizon

The BGP AS-Override Split-Horizon feature enables a Provider Edge (PE) device using split-horizon to avoid advertisement of routes propagated by a Customer Edge (CE) device to the same CE device. The BGP AS-Override Split-Horizon feature also enables a PE or CE device to send route updates to a specific PE or CE device in the same replication group.

Information About BGP AS-Override Split-Horizon

BGP AS-Override Split-Horizon Overview

When you configure split-horizon on a device, the Provider Edge (PE) device may advertise routes propagated from a Customer Edge (CE) device to the same CE device. The BGP AS-Override Split Horizon feature groups all the BGP neighbors into separate replication-groups, even when they are in the same update-group, and ensures that the route updates propagated from a CE device are not sent to the same CE device.

The BGP AS-Override Split Horizon feature enables a PE or CE device to selectively send and block updates to one or more neighboring PE or CE devices in the same update-group. The PE or CE device sends or blocks a message to a neighboring PE or CE device based on the type of the message and on whether the originator of the message matches the router ID of the PE or CE device.

How to Configure BGP AS-Override Split-Horizon

Configuring BGP AS-Override Split-Horizon

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address family ipv4 vrf** *vrf-name*
5. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
6. **neighbor** *ip-address* **activate**
7. **neighbor** *ip-address* **as-override split-horizon**
8. Repeat Step 5 to Step 7 to enable split-horizon for different neighbors in a virtual routing and forwarding (VRF) instance.
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 21`	Configures the Border Gateway Protocol (BGP) routing process and enters router configuration mode.
Step 4	**address family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf vrf1`	Specifies the name of the VPN routing and forwarding (VRF) instance to associate with subsequent IPv4 address family configuration mode commands and enters address-family configuration mode.
Step 5	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 192.0.2.1 remote-as 1`	Configures peering with a BGP neighbor in the specified autonomous system.
Step 6	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 192.0.2.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 address family with the local device.
Step 7	**neighbor** *ip-address* **as-override split-horizon** **Example:**	Enables split-horizon per neighbor in a VRF instance.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 192.0.2.1 as-override split-horizon`	
Step 8	Repeat Step 5 to Step 7 to enable split-horizon for different neighbors in a virtual routing and forwarding (VRF) instance.	—
Step 9	**end** **Example:** `Device(config-router-af)# end`	Exits router address-family configuration mode and enters privileged EXEC mode.

Verifying BGP AS-Override Split-Horizon

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpn4 all update-group**
3. **show ip bgp vpnv4 all neighbors** *ip-address*
4. **show ip bgp vpnv4 all neighbors** *ip-address* **policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpn4 all update-group** **Example:** `Device# show ip bgp vpn4 all update-group`	Displays information on update groups.
Step 3	**show ip bgp vpnv4 all neighbors** *ip-address* **Example:** `Device# show ip bgp vpnv4 all neighbors 192.0.2.1`	Displays details about neighbor connections.
Step 4	**show ip bgp vpnv4 all neighbors** *ip-address* **policy** **Example:** `Device# show ip bgp vpnv4 all neighbors 192.0.2.1 policy`	Displays neighbor policies per address-family.

Configuration Examples for BGP AS-Override Split-Horizon

Example: BGP AS-Override Split-Horizon Configuration

```
Device> enable
Device# configure terminal
Device(config)# router bgp 21
Device(config-router)# address-family ipv4 vrf vrf1
Device(config-router-af)# neighbor 192.0.2.1 remote-as 1
Device(config-router-af)# neighbor 192.0.2.1 activate
Device(config-router-af)# neighbor 192.0.2.1 as-override split-horizon
Device(config-router-af)# neighbor 198.51.100.1 remote-as 1
Device(config-router-af)# neighbor 198.51.100.1 activate
Device(config-router-af)# neighbor 198.51.100.1 as-override split-horizon
Device(config-router-af)# end
```

Example: Verifying BGP AS-Override Split-Horizon

Sample output for the show ip bgp vpn4 all update-group command

To display information about update groups, use the **show ip bgp vpn4 all update-group** command in privileged EXEC mode.

```
Device> enable
Device# show ip bgp vpn4 all update-group
BGP version 4 update-group 3, external, Address Family: VPNv4 Unicast
  BGP Update version : 5/0, messages 0 active RGs: 2 <<<<<<<<<<<<<<
  Overrides the neighbor AS 1 with my AS before sending updates
  Topology: blue, highest version: 5, tail marker: 5
  Format state: Current working (OK, last not in list)
                Refresh blocked (not in list, last not in list)
  Update messages formatted 1, replicated 2, current 0, refresh 0, limit 1000
  Number of NLRIs in the update sent: max 4, min 0
  Minimum time between advertisement runs is 0 seconds
  Has 2 members:
   192.0.2.1         198.51.100.1
```

Sample output for the show ip bgp vpnv4 all neighbors *ip-address* command

To display details about neighbor connections, use the **show ip bgp vpnv4 all neighbors** *ip-address* command in privileged EXEC mode.

```
Device> enable
Device# show ip bgp vpnv4 all neighbors 209.165.200.228
BGP neighbor is 209.165.200.228,  vrf vrf1,  remote AS 1, external link
  BGP version 4, remote router ID 209.165.201.28
  BGP state = Established, up for 00:01:26
  Last read 00:00:35, last write 00:00:28, hold time is 180, keepalive interval is 60 seconds

  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
```

```
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                6          2
    Keepalives:             3          3
    Route Refresh:          0          0
    Total:                 12          6
  Default minimum time between advertisement runs is 0 seconds

 For address family: VPNv4 Unicast
  Translates address family IPv4 Unicast for VRF vrf1
  Session: 209.165.200.228
  BGP table version 40, neighbor version 40/0
  Output queue size : 0
  Index 1, Advertise bit 1
  1 update-group member
  Overrides the neighbor AS with my AS before sending updates
  Split horizon processing before sending updates
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:            10          2 (Consumes 160 bytes)
    Prefixes Total:              10          2
    Implicit Withdraw:            0          0
    Explicit Withdraw:            0          0
    Used as bestpath:           n/a          2
    Used as multipath:          n/a          0
Outbound    Inbound
  Local Policy Denied Prefixes:    --------    -------
    Total:                                0          0
  Number of NLRIs in the update sent: max 5, min 0
  Last detected as dynamic slow peer: never
  Dynamic slow peer recovered: never
  Refresh Epoch: 1
  Last Sent Refresh Start-of-rib: 00:01:26
  Last Sent Refresh End-of-rib: 00:01:26
  Refresh-Out took 0 seconds
  Last Received Refresh Start-of-rib: never
  Last Received Refresh End-of-rib: never
                                     Sent       Rcvd
        Refresh activity:            ----       ----
          Refresh Start-of-RIB        1          0
          Refresh End-of-RIB          1          0

  Address tracking is enabled, the RIB does have a route to 209.165.200.228
  Connections established 3; dropped 2
  Last reset 00:01:35, due to split-horizon config change of session 1
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Mininum incoming TTL 0, Outgoing TTL 1
Local host: 209.165.200.225, Local port: 22789
Foreign host: 209.165.200.228, Foreign port: 179
Connection tableid (VRF): 2
```

Sample output for the show ip bgp vpnv4 all neighbors *ip-address* policy command

To display neighbor policies per address-family, use the **show ip bgp vpnv4 all neighbors** *ip-address* **policy** command in privileged EXEC mode.

```
Device> enable
Device# show ip bgp vpnv4 all neighbors 209.165.200.228
Neighbor: 209.165.200.228, Address-Family: VPNv4 Unicast (vrf1)
 Locally configured policies:
  as-override split-horizon
```

Additional References for BGP AS-Override Split-Horizon

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP AS-Override Split-Horizon

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 153: Feature Information for BGP AS-Override Split-Horizon

Feature Name	Releases	Feature Information
BGP AS-Override Split-Horizon		The BGP AS-Override Split-Horizon feature enables a Provider Edge (PE) device using split-horizon to avoid advertisement of routes propagated by a Customer Edge (CE) device to the same CE device. The BGP AS-Override Split-Horizon feature also enables a PE or CE device to send route updates to specific PE or CE device in the same replication group. The following command was introduced or modified: **neighbor** *ip-address* **as-override split-horizon**.

CHAPTER 124

BGP Support for Multiple Sourced Paths Per Redistributed Route

The BGP Support for Multiple Sourced Paths per Redistributed Route feature allows multiple paths with route redistribution or other sourcing mechanisms like the **network** command into BGP. This feature also allows multiple paths from the same source to be imported and exported across virtual routing and forwarding (VRF) instances.

This module provides an overview of the feature and describes how to configure it.

Restrictions for BGP Support for Multiple Sourced Paths Per Redistributed Route

The following restriction apply to this feature:

- Paths that are sourced with 0.0.0.0 as the gateway address will not have multiple paths redistributed into the Border Gateway Protocol (BGP). As a result, every sourced path must have a unique gateway.
- Suppose that for a prefix in the RIB we have a manually configured static route with a tag and a route without a tag inserted through RRI. In such a scenario, the route selection may be inconsistent, and either the manually configured route or the RRI route may be chosen.

  To prevent such an inconsistency, perform one of the following actions:

  - If you are manually configuring static routes to all the peer VPN networks of the router, disable RRI by removing reverse route configuration from the crypto map.
  - Set an identical tag in the crypto map for the route inserted through RRI.

Information About BGP Support for Multiple Sourced Paths Per Redistributed Route

BGP Support for Multiple Sourced Paths Per Redistributed Route Overview

The BGP Support for Multiple Sourced Paths per Redistributed Route feature allows multiple paths with route redistribution or other sourcing mechanisms like the **network** command into the Border Gateway Protocol (BGP). Prior to this feature, BGP accepted only one path from the Routing Information Base (RIB) to create a single BGP-sourced path for a redistributed network; even if the RIB had more than one path for the same network.

This feature also allows multiple paths from the same source to be imported and exported across virtual routing and forwarding (VRF) instances. Import of more than the default path into a VRF instance was already supported in BGP. However, these multiple paths had to be from different neighbors or sources and not from the same source.

By enabling this feature, customers can export Equal Cost Multipath (ECMP) sourced paths or next-hops from one VRF into hundreds of VRFs on the same device using BGP. Each of these paths are installed as multipaths into the RIB, and provides ECMP paths in other VRFs also.

For BGP to accept all the paths or next-hops per route from the redistributing protocol in the RIB, configure the **bgp sourced-paths** command. If you either disable or do not enable this command, BGP allows the import of only one sourced path per network from the RIB.

How to Configure BGP Support for Multiple Sourced Paths Per Redistributed Routes

Configuring Multiple Sourced Paths

When you configure the **bgp sourced-paths** command, the Border Gateway Protocol (BGP) accepts all paths from the Routing Information Base (RIB). When the **bgp sourced-paths** command is removed, the configuration returns to the default behavior of allowing only one sourced path per network from the RIB into BGP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4 vrf** *vrf-name*
5. **bgp sourced-paths per-net static all**
6. **redistribute static**
7. **neighbor** *ip-address* **remote-as** *neighbor-as*
8. **neighbor** *ip-address* **activate**
9. **neighbor** *ip-address* **send-community both**

10. **end**

DETAILED STEPS

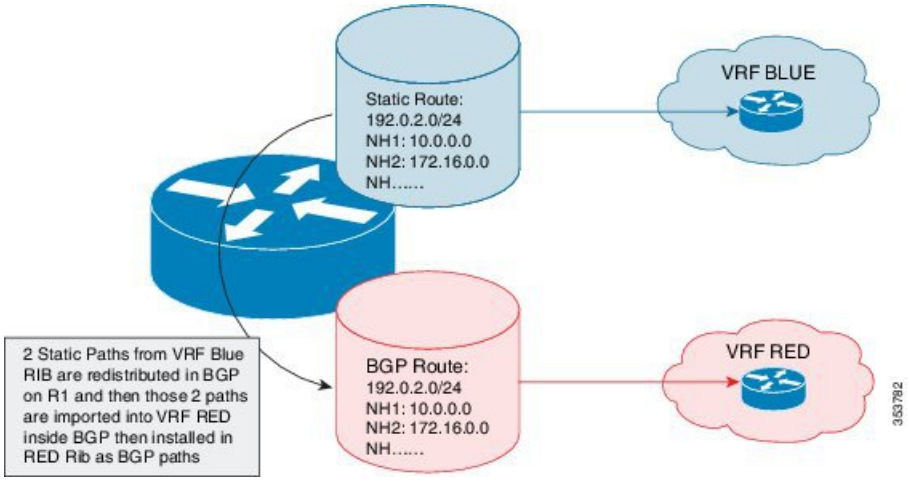
	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 65000`	Configures the BGP routing process and enters the routing configuration mode.
Step 4	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf blue`	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes. **Note** You can also configure the **address-family ipv6** command based on your network configuration.
Step 5	**bgp sourced-paths per-net static all** **Example:** `Device(config-router-af)# bgp sourced-paths per-net static all`	Allows per network sourcing of all static paths in the RIB.
Step 6	**redistribute static** **Example:** `Device(config-router-af)# redistribute static`	Redistributes static routes from another routing protocol.
Step 7	**neighbor** *ip-address* **remote-as** *neighbor-as* **Example:** `Device(config-router-af)# neighbor 204.0.0.3 remote-as 65000`	Adds an entry into the BGP or multiprotocol BGP neighbor table.
Step 8	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 204.0.0.3 activate`	Enables the exchange of information with a BGP neighbor.
Step 9	**neighbor** *ip-address* **send-community both** **Example:** `Device(config-router-af)# neighbor 204.0.0.3 send-community both`	Specifies that a communities attribute should be sent to a BGP neighbor.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for BGP Multiple Sourced Paths Per Redistributed Route

Example: Configuring Multiple Sourced Paths

Figure 129: Deployment Scenario for BGP Multiple Paths Replication



The above figure displays a deployment scenario in which BGP replicates multiple paths from VRF BLUE to VRF RED. VRF RED can import more paths, in addition to the best-path, by using the same route target export in VRF BLUE and VRF RED. This helps multiple paths to get into VRF RED.

```
Device# configure terminal
Device(config)# ip vrf blue
Device(config-vrf)# rd 100:200
Device(config-vrf)# route-target export 200:200
Device(config-vrf)# route-target import 200:200
Device(config-vrf)# exit

Device(config)# ip vrf red
Device(config-vrf)# rd 200:200
Device(config-vrf)# route-target export 300:200
Device(config-vrf)# route-target import 300:200
Device(config-vrf)# route-target import 200:200
Device(config-vrf)# exit

Device(config)# interface Loopback 0
Device(config-if)# ip address 198.51.100.1 255.255.255.255
```

```
Device(config-if)# exit

Device(config)# interface Ethernet 1/0
Device(config-if)# ip address 203.0.113.1 19.0.0.32 255.255.255.255
Device(config-if)# no shutdown
Device(config-if)# exit

Device(config)# interface Ethernet 1/2
Device(config-if)# ip address 209.165.200.225 255.255.255.240
Device(config-if)# no shutdown
Device(config-if)# exit

Device(config)# interface Ethernet 1/2.2
Device(config-subif)# encapsulation dot1Q 2
Device(config-subif)# ip vrf forwarding blue
Device(config-subif)# ip address 192.168.0.1 255.255.255.240
Device(config-subif)# no shutdown
Device(config-subif)# exit

Device(config)# interface Ethernet 1/2.3
Device(config-subif)# encapsulation dot1Q 3
Device(config-subif)# ip vrf forwarding blue
Device(config-subif)# ip address 1922.168.0.17 255.255.255.240
Device(config-subif)# no shutdown
Device(config-subif)# exit

Device(config)# router ospf 2 vrf blue
Device(config-router)# network 192.68.0.0 0.0.0.255 area 0
Device(config-router)# network 192.68.1.16 0.0.0.255 area 0
Device(config-router)# exit
!
Device(config)# router ospf 1
Device(config-router)# network 209.165.200.224 0.0.255.255 area 0
Device(config-router)# exit

Device(config)# router bgp 65000
Device(config-router)# no bgp default ipv4-unicast
Device(config-router)# neighbor 10.0.0.2 remote-as 65000
Device(config-router)# neighbor 10.0.0.2 update-source Loopback0
Device(config-router)# address-family ipv4
Device(config-router-af)# exit-address-family

Device(config-router)# address-family vpnv4
Device(config-router-af)# neighbor 10.0.0.2 activate
Device(config-router-af)# neighbor 10.0.0.2 send-community extended
Device(config-router-af)# exit-address-family

Device(config-router)# address-family ipv4 vrf blue
Device(config-router-af)# bgp sourced-paths per-net static all
Device(config-router-af)# bgp sourced-paths per-net ospf all
Device(config-router-af)# redistribute static
Device(config-router-af)# redistribute ospf 2
Device(config-router-af)# exit-address-family

Device(config-router)# address-family ipv4 vrf red
Device(config-router-af)# import path selection all
Device(config-router-af)# import path limit 2
Device(config-router-af)# maximum-paths 2
Device(config-router-af)# exit-address-family
Device(config-router)# exit

Device(config)# ip route vrf blue 192.0.2.2 255.255.255.255 10.0.0.2 global
```

```
Device(config)# ip route vrf blue 192.0.2.2 255.255.255.255 172.16.0.2 global
Device(config)# end
```

Additional References for BGP Support for Multiple Sourced Paths Per Redistributed Route

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for BGP Support for Multiple Sourced Paths Per Redistributed Route

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 154: Feature Information for BGP Support for Multiple Sourced Paths Per Redistributed Route

Feature Name	Releases	Feature Information
BGP Support for Multiple Sourced Paths Per Redistributed Route	Cisco IOS XE Release 3.15S	The BGP Support for Multiple Sourced Paths per Redistributed Route feature allows multiple paths with route redistribution or other sourcing mechanisms like the **network** command into BGP. This feature also allows multiple paths from the same source to be imported and exported across virtual routing and forwarding (VRF) instances. In Cisco IOS XE Release 3.15S, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced: **bgp sourced-paths**.

CHAPTER 125

Maintenance Function: BGP Routing Protocol

From Cisco IOS XE Everest 16.4.1 release, the event trace functionality is supported for BGP. Event Trace provides the functionality to capture BGP traces by enabling the event trace using commands. You can disable the command if you do not want to log traces. When convergence happens and connection states are getting changed, the BGP traces are logged into Event Trace infrastructure.

Information About Maintenance Function: BGP Routing Protocol

BGP Event trace supports the following functionalities:

- BGP Event Trace creates buffers for peer connection state change and updates event logging. The size of the buffer is 100,000, which means 100,000 trace entries will be stored at a time. The buffer can be resized by using configuration command and maximum size of the buffer can be extended till 1,000,000.
- These buffers are circular in nature, that is, if the buffer reaches the end then it starts logging from the beginning. If "one-shot" is not configured, it continuously logs from the beginning.
- Considering the contribution is a small addition to performance, BGP event trace will be disabled by default. It can be enabled by executing the **enable** command in EXEC mode.
- BGP Event Traces:
  - Neighbor: All the peer events such as state changes, error handling, unrecognized/malformed packet handling will be captured into this buffer.
- BGP logs the traces in binary format into corresponding buffers on runtime, which helps in logging the trace efficiently. Use the **monitor event-trace bgp neighbor** command to print the traces in human-readable format on the console. This command provides the functionality of dumping the event traces into the file in binary or human-readable format as well.
- The show commands are provided with afi/safi/vrf/neighbor address filtering options to display the event logs. Event Trace logging under different afi/safi/vrf is completely based on the different traces.

Configuring BGP Event Trace in Global Configuration Mode

BGP Event Trace provides the commands in global configuration and privileged EXEC mode for connection state event traces. Use the following configuration steps to enable the event-traces for BGP. With this configuration, BGP traces are enabled after the active/standby router is rebooted because of a crash or switchover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **monitor event-trace bgp neighbor** {**dump-file** *filename* | **size** *entries*}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**monitor event-trace bgp neighbor** {**dump-file** *filename*	**size** *entries*} **Example:** `Device(config)# monitor event-trace bgp neighbor size 10`	Enables event traces for BGP. Use the **no monitor event-trace bgp neighbor** command to disable the event traces. • **dump-file**—Set the name of the trace dump file. • **size**—Set the size of trace.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Configuring BGP Event Trace in EXEC Mode

SUMMARY STEPS

1. **enable**

2. **monitor event-trace bgp neighbor** {**clear** | **continuous** | **destroy-buffer** | **disable** | **dump** *filename* | **enable** | **one-shot**}
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor event-trace bgp neighbor** {**clear**	**continuous**	**destroy-buffer**	**disable**	**dump** *filename*	**enable**	**one-shot**} **Example:** `Device# monitor event-trace bgp neighbor enable`	Enables event traces for BGP. Use the **no monitor event-trace bgp neighbor** command to disable the event traces. • **clear**--Clear the event trace buffer. • **continuous**--Display the event traces getting logged continuously on the console. • **destroy-buffer**--Destroy buffer allocated for traces. • **disable**--Disable the event trace functionality. This command must be given on active and standby to disable both nodes. • **dump** *filename*--Dump all neighbor event traces into file in binary or ASCII format. • **enable**--Enable the event trace functionality. This command must be given on active and standby to enable both nodes. • **one-shot**--Log the event trace only once. When the buffer is full, the event-trace logging will stop.
Step 3	**exit** **Example:** `Device# exit`	Exits the privileged EXEC configuration mode.

Verifying the BGP Event Traces

You can use the following **show** commands to browse through the event traces captured. These **show** commands will filter the traces based on the AFI/SAFI/VRF/neighbor address and different combinations.

- **show monitor event-trace bgp all**
- **show monitor event-trace bgp back**
- **show monitor event-trace bgp clock**
- **show monitor event-trace bgp from-boot**
- **show monitor event-trace bgp ipv4** {**all** | **back** | **clock** | **flowspec** | **from-boot** | **latest** | **mdt** | **multicast** | **mvpn** | **unicast**}

- **show monitor event-trace bgp ipv4 flowspec neighbors**
- **show monitor event-trace bgp ipv4 mdt vrf**
- **show monitor event-trace bgp ipv6** {**all** | **back** | **clock** | **flowspec** | **from-boot** | **latest** | **multicast** | **mvpn** | **unicast**}
- **show monitor event-trace bgp l2vpn** {**all** | **back** | **clock** | **evpn** | **from-boot** | **latest** | **vpls**}
- **show monitor event-trace bgp latest**
- **show monitor event-trace bgp neighbors**
- **show monitor event-trace bgp nsap**
- **show monitor event-trace bgp parameters**
- **show monitor event-trace bgp rtfilter**
- **show monitor event-trace bgp vpnv4** {**all** | **back** | **clock** | **from-boot** | **latest** | **vrf**}
- **show monitor event-trace bgp vpnv6** {**all** | **back** | **clock** | **flowspec** | **from-boot** | **latest** | **multicast** | **unicast**}

Feature Information for Maintenance Function: BGP Routing Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 155: Feature Information for Maintenance Function: BGP Routing Protocol

Feature Name	Releases	Feature Information
Maintenance Function: BGP Routing Protocol	Cisco IOS XE Everest 16.4.1 Release	From Cisco IOS XE Everest 16.4.1 release, the event trace functionality is supported for BGP. Event Trace provides the functionality to capture BGP traces by enabling the event trace using commands. You can disable the command if you do not want to log traces. When convergence happens and connection states are getting changed, the BGP traces are logged into Event Trace infrastructure.

CHAPTER 126

BGP Support for TCP Authentication Option

This document describes how to configure Message Digest5 (MD5) authentication on a Transmission Control Protocol (TCP) connection between two BGP peers.

BGP Support for TCP AO Overview

On a secure control plane, BGP uses Message Digest 5 (MD5) algorithm as the authentication mechanism. It uses the TCP API to configure the keychain on a TCP connection. When authentication is enabled, any Transmission Control Protocol (TCP) segments belonging to BGP are exchanged between peers, verified and then accepted only if authentication is successful. BGP application use the TCP API to configure the keychain on a TCP connection. It owns the configuration to associate a TCP-AO keychain name with a neighbor, a peer-group, or a peer-session template.

You can validate the authentication configuration per neighbor/peer-group/peer-session template. Authentication Option is supported for BGP dynamic neighbor, BGP Non-stop forwarding (NSF) and Non-stop routing (NSR). Routing protocols support a different set of cryptographic algorithms, however, BGP supports only MD5. For example, if BGP is configured with the TCP MD5 key (md5-key), it will not allow to configure TCP-AO and vice versa.There are two options to configure BGP:

- **include-tcp-options** - option to specify if the TCP option headers (other than TCP AO option) will be included while computing the MAC digest of the packets.
- **accept-ao-mismatch-connections** - option to accept the connection as non-TCP AO connection when receives a connection from peer without TCP AO option. Similarly, if the connection is initiated from one side, the peer acknowledges with TCP AO, it accepts the ACK and continues the connection.

Restrictions

- Configuring and deconfiguring TCP AO for a certain neighbor or peer-group or peer-session causes existing established BGP session(s) to flap.
- Do not change the configuration of an existing TCP key chain because existing BGP sessions may break.

- TCP OA must be configured between peers with compatible versions, either both running 17.6.2 or later, or both running releases earlier than 17.6.2.
- TCP AO picks up the most valid key under the key chain. The most valid key is the one which has the longest send lifetime. If there are two keys with the same send lifetime, the first best key is selected.
- In a configuration, where one of the devices is configured with the TCP MD5 option and the other with the TCP-AO option not supported, BGP session is not established between the devices until you correct the configuration.
- After a session is established using a specific key chain, if you modify the key chain, the session ends, and an attempt is made to renegotiate the session based on the modified key chain.

How to Configure BGP Using TCP AO

The BGP application must be configured on both the devices. To establish a peer connection with TCP-AO, you must configure the following:

Configuring TCP Key Chain and Keys

Before you begin

TCP-AO key chain and keys must be configured on both the peers communicating through a TCP connection.

- Ensure that the key-string, send-lifetimes, and ids of keys match on both peers.
- Ensure that the send-id on a router matches the receiver-id on the peer router. Also, use the same ID for the parameters.
- The send-id and recv-id of a key cannot be reused for another key in the same key chain.
- Do not modify a key that is in use. Disassociate the key from the TCP connection before modifying the key.

Step 1 enable

Example:

```
Router# enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 configure terminal

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 key chain *key-chain-name* tcp

Example:

```
Router(config)# key chain kc1 tcp
```

Creates a TCP-AO key chain with the specified name and enters the TCP-AO key chain configuration mode.

The key chain name can have a maximum of 256 characters.

Step 4 key *key-id*

Example:

```
Router(config-keychain-tcp)# key 10
```

Creates a key with the specified key-id and enters the the TCP-AO key chain key configuration mode.

The key-id must be in the range 0 to 2147483647.

Step 5 send-id *send-identifier*

Example:

```
Router(config-keychain-tcp-key)# send-id 218
```

Specifies the send identifier for the key.

The send-identifier must be in the range 0 to 255.

Step 6 recv-id *receive-identifier*

Example:

```
Router(config-keychain-tcp-key)# recv-id 218
```

Specifies the receive identifier for the key.

The receive-identifier must be in the range 0 to 255.

Step 7 cryptographic-algorithm {aes-128-cmac | hmac-sha-1 | hmac-sha-256}

Example:

```
Router(config-keychain-tcp-key)# cryptographic-algorithm hmac-sha-1
```

Specifies the algorithm to be used to compute MACs for TCP segments.

Step 8 include-tcp-options

Example:

```
Router(config-keychain-tcp-key)# include-tcp-options
```

(Optional)

This flag indicates whether TCP options other than TCP-AO must be used to calculate MACs.

With the flag enabled, the content of all options, in the order present, is included in the MAC and TCP-AO's MAC field is filled with zeroes.

When the flag is disabled, all options other than TCP-AO are excluded from MAC calculations.

This flag is disabled by default.

Step 9 send-lifetime [local] *start-time duration*

Example:

```
Router(config-keychain-tcp-key)# send-lifetime local 12:00:00 28 Feb 2018 duration 20
```

Specifies the time for which the key is valid to be used for TCP-AO authentication.

Use the local keyword to specify the start-time in the local time zone. By default, the start-time corresponds to UTC time.

Step 10 accept-lifetime [local] *start-time duration*

Example:

```
Router(config-keychain-tcp-key)# accept-lifetime local 12:00:00 28 Feb 2018 duration 20
```

Specifies the time for which the key is valid to be used for TCP-AO authentication.

Use the local keyword to specify the start-time in the local time zone. By default, the start-time corresponds to UTC time.

Step 11 key-string *master-key*

Example:

```
Router(config-keychain-tcp-key)# key-string master-key
```

Specifies the master-key for deriving traffic keys.

The master-key must have 32 or 64 hexadecimal digits.

The master-keys must be identical on both the peers. If the master-keys do not match, authentication fails and segments may be rejected by the receiver.

Step 12 accept-ao-mismatch

Example:

```
Router(config-keychain-tcp-key)# accept-ao-mismatch
```

(Optional) This flag indicates whether the receiver should accept segments for which the MAC in the incoming TCP AO does not match the MAC generated on the receiver.

Step 13 end

Example:

```
Router(config-keychain-tcp-key)# end
```

Exits TCP-AO key chain key configuration mode and returns to privileged EXEC mode.

Example

A simple key chain configuration show on 2 end points A and B of the TCP AO enabled connection:

```
R1:

key chain kc1 tcp
key 7890
  send-id 215
  recv-id 215
  key-string klomn
  cryptographic-algorithm hmac-sha-1
 include-tcp-options

R2:

key chain kc1 tcp
key 7890
```

```
send-id 215
recv-id 215
key-string klomn
cryptographic-algorithm hmac-sha-1
include-tcp-options
```

Configuring BGP Peer- group and Peer-session

BGP neighbor configuration

To configure a BGP neighbor using TCP AO:

```
Router(config)# router bgp <own-AS>
Router(config-router)# neighbor <peer-IP-address|peer-IPv6-address> ao <keychain-name>
[include-tcp-options] [accept-ao-mismatch-connections]
```

You can also configure BGP dynamic neighbor using the above command. Use the no form of the commands to deconfigure BGP neighbor.

BGP peer-group configuration

To configure a BGP peer-group using TCP AO:

```
Router(config-router)# neighbor <peer-group-name> ao <keychain-name>
[include-tcp-options] [accept-ao-mismatch-connections]
```

You can also configure BGP dynamic neighbor using the above command. Use the no form of the commands to deconfigure BGP peer-group.

BGP peer-session configuration

To configure a BGP peer-session template using TCP AO:

```
Router(config-router)# template peer-session <session-name>
Router(config-router-stmp)# ao <keychain-name> [include-tcp-options]
[accept-ao-mismatch-connections]
```

Use the no form of the command to deconfigure BGP peer-session.

Verifying TCP-AO Key Chain and Key Configuration

Use the **show key chain key-chain-name** command in the privileged exec mode to display information about a TCP-AO key chain and keys.

```
Router# show key chain key-chain-name
```

```
Router1# show key chain kc2
Key-chain kc2:
TCP key chain
key 7893 -- text "abcde"
cryptographic-algorithm: hmac-sha-1
accept lifetime (12:32:00 IST Nov 9 2018) - (10:30:00 IST Dec 30 2019) [valid now]
send lifetime (13:05:00 IST Jan 12 2019) - (10:31:00 IST Dec 30 2019) [valid now]
send-id - 218
recv-id - 218
include-tcp-options
MKT ready - true
```

```
MKT preferred - true
MKT in-use - true
MKT id - 7893
MKT send-id - 218
MKT recv-id - 218
MKT alive (send) - true
MKT alive (recv) - true
MKT include TCP options - true
MKT accept AO mismatch - false
TCB - 0x7FBD68361838
curr key - 7893
next key - 7893
```

Verifying TCP-AO Key Chain Information in the TCB

Use the **show tcp tcb address-of-tcb** command in the priviledged exec mode to display information about TCP-AO in the Transmission Control Block. Obtain address-of-tcb (the hexadecimal address of the TCB) from the output of the **show key chain key-chain-name** command.

```
Router# show tcp tcb address-of-tcb

Router1# sh tcp tcb 7FBD68361838
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 1.0.2.1, Local port: 40125
Foreign host: 1.0.2.2, Foreign port: 5555
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x2818B07):
Timer          Starts    Wakeups            Next
Retrans             1          0             0x0
TimeWait            0          0             0x0
AckHold             1          0             0x0
SendWnd             0          0             0x0
KeepAlive        6651          0       0x281AC36
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
Linger              0          0             0x0
ProcessQ            0          0             0x0

iss: 3307331702  snduna: 3307331703  sndnxt: 3307331703
irs:  725047078  rcvnxt:  725047079

sndwnd:   4128  scale:      0  maxrcvwnd:   4128
rcvwnd:   4128  scale:      0  delrcvwnd:      0

SRTT: 125 ms, RTTO: 2625 ms, RTV: 2500 ms, KRTT: 0 ms
minRTT: 15 ms, maxRTT: 1000 ms, ACK hold: 200 ms
uptime: 40996359 ms, Sent idletime: 6505 ms, Receive idletime: 6505 ms
Status Flags: active open
Option Flags: keepalive running, nagle, Retrans timeout
IP Precedence value : 0

TCP AO Key chain: kc2

TCP AO Current Key:
```

```
 Id: 7893, Send-Id: 218, Recv-Id: 218
 Include TCP Options: Yes*
 Accept AO Mismatch: No*

TCP AO Next Key:
 Id: 7893, Send-Id: 218, Recv-Id: 218
 Include TCP Options: Yes*
 Accept AO Mismatch: No*

Datagrams (max data segment is 1460 bytes):
Rcvd: 4372 (out of order: 0), with data: 0, total data bytes: 0
Sent: 4372 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with
data: 0, total data bytes: 0

 Packets received in fast path: 0, fast processed: 0, slow path: 0
 fast lock acquisition failures: 0, slow path: 0
TCP Semaphore      0x7FBD6801B2E0  FREE

* - Derived from Key
```

Example: Verifying BGP Configuration

Use the **show ip bgp peer-group peer-group-name** and **show ip bgp template peer-session peer-session-name** commands in the privileged exec mode to display information about BGP configuration on peer groups and peer sessions.

```
Router# show ip bgp peer-group ABC

BGP peer-group is ABC,  remote AS 100
  BGP version 4
  …
  AO keychain <keychain-name> include-tcp-options accept-ao-mismatch-connections
  …



Router# show ip bgp template peer-session ABC

Template:ABC,  index:1
Local policies:<hex-value>, Inherited polices:<hex-value>
Locally configured session commands:
  …
  AO keychain <keychain-name> include-tcp-options accept-ao-mismatch-connections
  …
Inherited session commands:
  …
  AO keychain <keychain-name> include-tcp-options accept-ao-mismatch-connections
```

CHAPTER 127

BGP Unlabeled and Labeled Unicast in the Same Session: Label-Unicast Unique Mode

Cisco IOS XE provides a mode called "label-unicast unique". With this mode enabled, in BGP sessions that include unlabeled unicast and labeled unicast, the unlabeled and labeled forms of a given prefix are treated as unique. This improves interoperability with other operating systems that also treat these as unique, such as Cisco IOS XR.

Overview

Devices operating with the Border Gateway Protocol (BGP) are termed BGP speakers. They maintain a Routing Information Base (RIB) that stores routing information for other BGP speakers in the network. The RIB includes a separate “Adjacent Routing Information Base, Incoming” (Adj-RIB-In) component for each BGP speaker, storing routing information received from that specific speaker.

For a given BGP speaker, neighboring speakers are considered peers. Typically, a speaker has only one peering relationship with another BGP speaker, but in some cases it may have multiple peering relationships with another BGP speaker. Typically, unless the multi-session feature is used, only one session is established for each peering relationship.

When a BGP speaker advertises unicast routes to its peers, it sends an UPDATE message. The routes include a prefix and possibly a label.

- Unlabeled unicast (uses SAFI 1): Unicast prefix only.
- Labeled unicast (uses SAFI 4): Unicast prefix, and an MPLS label associated to that prefix.

Note SAFI = Subsequent Address Family Identifier

Unlabeled and Labeled Unicast in the Same Session

In a single session, two BGP speakers can use both unlabeled and labeled unicast. Different BGP implementations handle this differently. When keeping track of the latest route information for a given prefix:

- **(a)** Some systems treat all unlabeled or labeled forms of a prefix as **equivalent**. These systems store the route information provided by the most recent update for the prefix, whether unlabeled or labeled.
- **(b)** Other systems treat each form of the prefix, whether unlabeled (SAFI 1) or labeled (SAFI 4), as **unique**. With these systems, the Adj-RIB-In for each peer stores routing information separately for the unlabeled prefix and the labeled prefix.

For a BGP session to propagate routes correctly, the way that one speaker sends a prefix must match the way that the other side receives the prefix. If they are not matched, the routes will not be received correctly. In that case, the receiver will then install the routes incorrectly, but without indicating any error, so the problem may go unaddressed.

Before Cisco IOS XE Gibraltar 16.12.1, IOS XE used only the method described in (a) above. It supported unlabeled and labeled unicast in the same session, but only among Cisco IOS XE speakers. Any two forms of the same prefix, whether unlabeled or labeled, were considered **equivalent**. For a given peer, the last received version of route information for any specific prefix was stored in the Adj-RIB-In, regardless of whether it was unlabeled or labeled.

Beginning with release 16.12.1, Cisco IOS XE provides a mode called "label-unicast unique", which supports sessions with BGP speakers that handle each form (unlabeled or labeled) of a prefix as **unique**. In this mode, the Adj-RIB-In for a given peer can store two routes for the same prefix, one route for unlabeled (received as SAFI 1) and one route for labeled (received as SAFI 4).

In this mode:

- Inbound processing: The device can store one labeled path and one unlabeled path from a neighbor.
- Outbound processing: The device executes update-generation twice, once for SAFI 1 and once for SAFI 4. Prefixes are advertised and withdrawn independently for each SAFI. In most scenarios, a prefix is only advertised in one of the SAFI types, but this depends on policy and may not be true in all cases.

The following figures show some of the difference in behavior that "label-unicast unique" mode provides by storing route information separately for unlabeled and labeled forms of the same prefix.

Figure 130: Non-unique Mode: Unlabeled and Labeled Forms of a Prefix Considered Equivalent, Stores Only Last Route Received

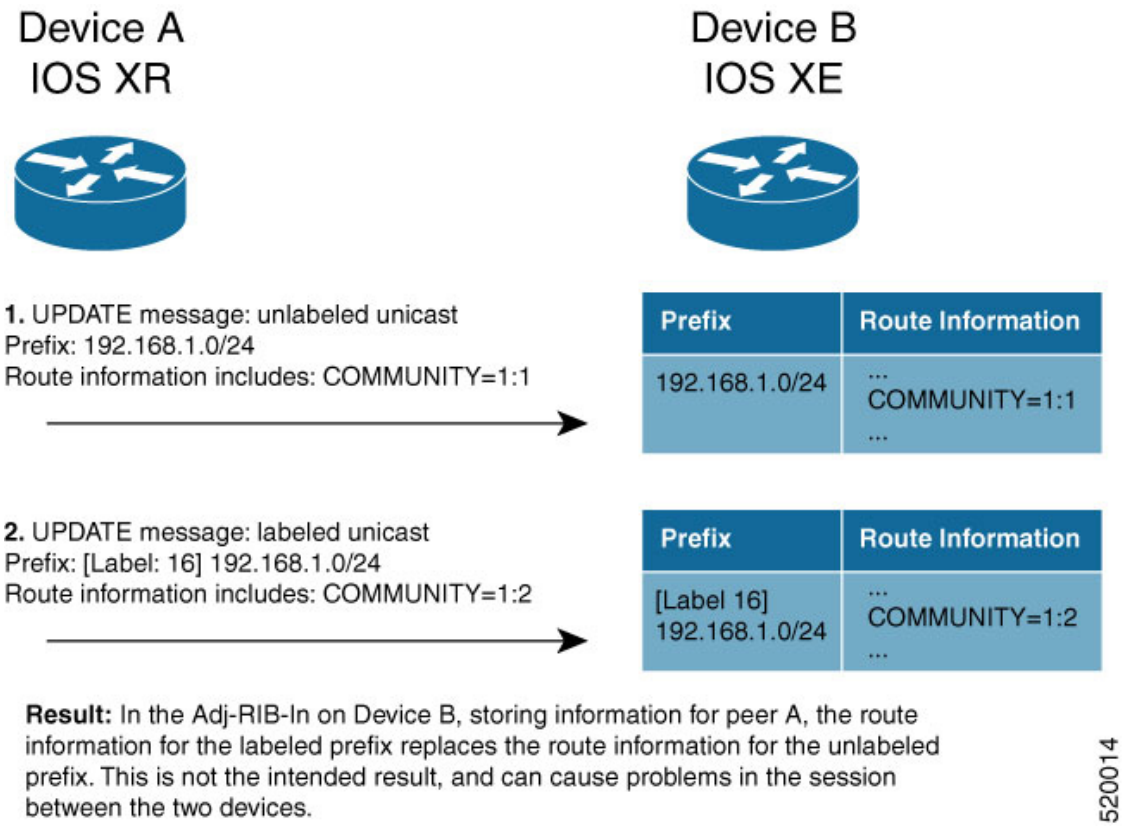
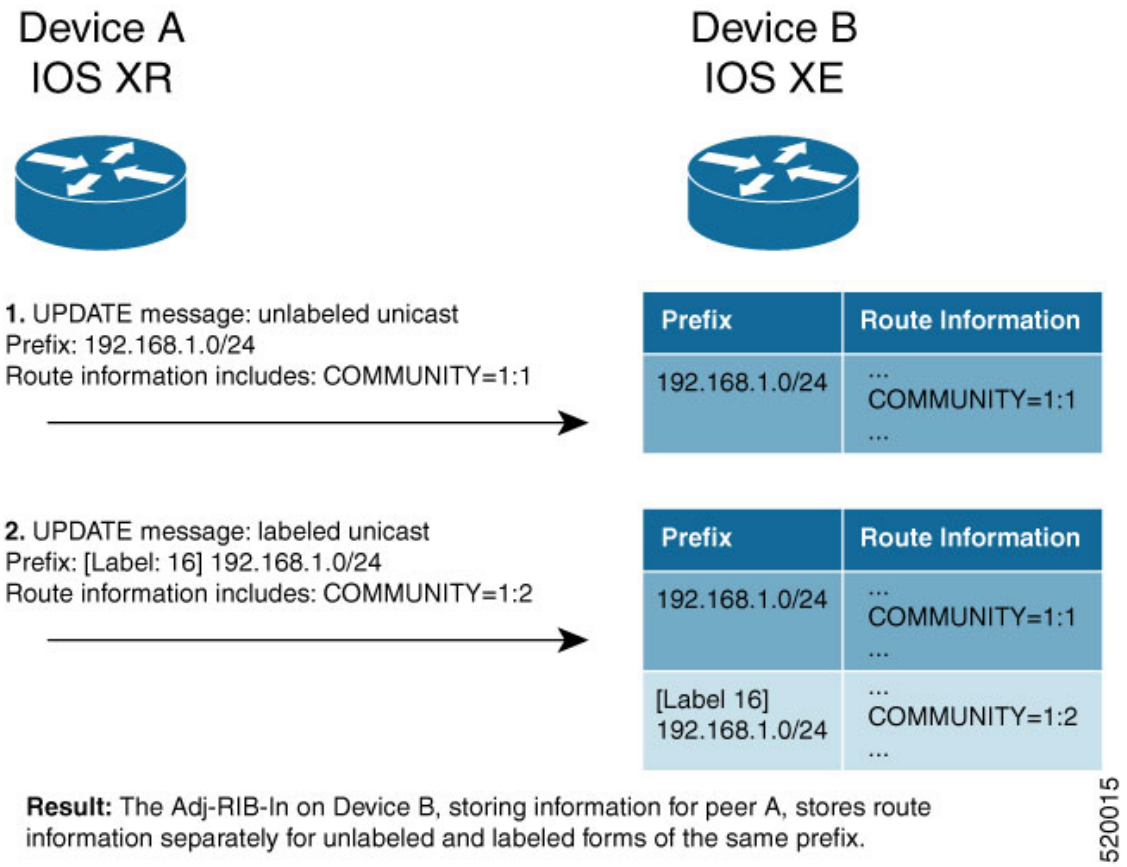


Figure 131: "Label-Unicast Unique" Mode: Unlabeled and Labeled Forms of a Prefix Considered Unique, Routes for Each Stored Separately



Interoperability

The "label-unicast unique" mode improves interoperability between Cisco IOS XE and operating systems, such as Cisco IOS XR, that treat unlabeled and labeled forms of a prefix as unique.

Backward Compatibility

To preserve backward compatibility, Cisco IOS XE Gibraltar 16.12.1 operates by default in the same mode as in previous releases. To change to the "label-unicast unique" mode, use the **update {in|out} labeled-unicast unique** command.

Restrictions

- BGP additional-paths is not supported with the label-unicast unique feature configured.

Configuration

Symmetrical Configuration

The term "symmetrical" applies to sessions between BGP speakers that both treat unlabeled and labeled prefixes as unique, for inbound (when receiving) and outbound (when advertising). This is the ideal scenario. An example would be a device using Cisco IOS XE communicating with a device using Cisco IOS XR.

See below for details of configuring symmetrical devices.

Two Cisco IOS XE Devices

In this configuration example:

- Device A: Using Cisco IOS XE. Configured with IP address 10.0.0.1, Autonomous System (AS) identifier 100.
- Device B: Using Cisco IOS XE. Configured with IP address 10.0.0.2, AS identifier 200.

On device A, in router configuration mode:

```
router bgp 100
neighbor 10.0.0.2 remote-as 200
neighbor 10.0.0.2 update in labeled-unicast unique
neighbor 10.0.0.2 update out labeled-unicast unique
address-family ipv4 unicast
neighbor 10.0.0.2 send-label
```

On device B, in router configuration mode:

```
router bgp 200
neighbor 10.0.0.1 remote-as 100
neighbor 10.0.0.1 update in labeled-unicast unique
neighbor 10.0.0.1 update out labeled-unicast unique
address-family ipv4 unicast
neighbor 10.0.0.1 send-label
```

One Cisco IOS XE Device and One Other Device

In this configuration example:

- Device A: Using Cisco IOS XE. Configured with IP address 10.0.0.1, Autonomous System (AS) identifier 100.
- Device B: Using Cisco IOS XR. Configured with IP address 10.0.0.2, AS identifier 200.

The configuration steps are required only on device A, running IOS XE.

On device A, in router configuration mode:

```
router bgp 100
neighbor 10.0.0.2 remote-as 200
neighbor 10.0.0.2 update in labeled-unicast unique
neighbor 10.0.0.2 update out labeled-unicast unique
address-family ipv4 unicast
neighbor 10.0.0.2 send-label
```

Asymmetrical Configuration

The term "asymmetrical" applies to sessions between BGP speakers, where one speaker treats unlabeled and labeled prefixes as unique for inbound and as equivalent for outbound; and the other speaker treats unlabeled and labeled prefixes as equivalent for inbound and unique for outbound.

This is not ideal, but arises in some network scenarios.

In this configuration example:

- Device A: Using Cisco IOS XE. Configured with IP address 10.0.0.1, Autonomous System (AS) identifier 100.
- Device B: Using Cisco IOS XE. Configured with IP address 10.0.0.2, AS identifier 200.

On device A, in router configuration mode:

```
router bgp 100
neighbor 10.0.0.2 remote-as 200
neighbor 10.0.0.2 update-labeled in unicast unique
address-family ipv4 unicast
neighbor 10.0.0.2 send-label
```

On device B, in router configuration mode:

```
router bgp 200
neighbor 10.0.0.1 remote-as 100
neighbor 10.0.0.1 update-labeled out unicast unique
address-family ipv4 unicast
neighbor 10.0.0.1 send-label
```

CHAPTER 128

BGP Replace ASNs in the AS Path

This chapter explains how BGP replaces ASNs in the route map.

Information about BGP Replace ASNs

This document describes how using the BGP policy, any of the configured Autonomous System Numbers (ASNs) are replaced with its own ASN. You can replace one or many ASNs using the AS_PATH attribute.

Loop prevention in BGP is based on the verification of AS numbers in the AS Path. If the receiving router sees its own AS number in the AS Path of the received BGP packet, the packet is dropped. However, this may cause issue in an inter-AS Hub and Spoke, such as a central Firewall location which must advertise more specific prefixes back to the spokes so the inter-AS crossing can happen twice (back and forth). The existing **set as-path** command under route-map provides functionality to prepend to the as-path and set tag. This solution expands the **set as-path** command to allow replace a sequence of ASNs to own or local AS.

Restrictions for BGP Replace ASNs in the AS Path

- BGP Replace ASNs in the AS Path feature supports only eBGP neighbors on a per AFI basis.
- When configuring BGP replace ASNs on iBGP neighbors, a warning message appears saying, the route-map configuration is taken (not failed), but the **`set as-path replace`** clause is ignored.
- The route-map with replace as-path functionality is applied to both inbound and outbound side of BGP neighbor. For inbound side, BGP replaces ASNs after the AS-PATH loop detection.
- The **set as-path replace** command only operates on AS_SEQUENCE and AS_CONFED_SEQUENCE, not AS_SET and AS_CONFED_SET attributes.
- Translate 4-byte AS number to 2 byte AS number before sending BGP messages to a BGP speaker.

- If confederation id is configured, when talking to a peer that is outside the confederation, use confederation ID to replace the ASNs. When talking to a peer that belong to the same confederation, use member-AS number to replace the ASNs.
- When the BGP Local-AS feature is configured, the configured Local-AS is used to replace the ASNs defined.
- The maximum length of a command is defined by PARSEBUF, which is 256 bytes. So, the max number of ASNs that are allowed in a single is 256 bytes.
- When multiple set as-path replace are configured, each entry is applied in a chain recursively, where the output of the current entry is the next entry in the chain.
- If you configure both set as-path prepend and set as-path replace, BGP processes set as-path replace first, and then set as-path prepend.

Configure BGP Replace ASNs in the AS Path

To replace a sequence of ASNs to its own AS, use the set as-path replace command:

```
Device(config-route-map)# set as-path replace {any | as-path-string}
any replaces each AS number in the AS path with the local AS number.
```

```
Device(config-route-map)# set as-path replace {any | as-path-string}
as-path-string is a sequence of AS numbers.
```

Configuration Examples for BGP Replace ASNs in the AS Path

The following example shows how to replace AS numbers in the AS path. In the example, AS-Path is "67 100 65533 5 78 89 6 5 28 100 9", and locally configured ASN is 900:

```
Device(config)#route-map test
R1(config-route-map)# set as-path replace 100
```

In this configuration, all occurrences of 100 in the AS-path are replaced with a local AS. The new AS-Path will be "67 900 65533 5 78 89 6 5 28 900 9".

```
Device(config)#route-map test
Device(config-route-map)# set as-path replace 5 78
```

In this configuration, all occurrences of AS sequence "5 78" in the AS-path replaces all the ASNs in the configured AS sequence with local AS. The new AS-Path will be "67 100 65533 900 900 89 6 5 28 100 9".

The following example configures to replace multiple individual ASNs or AS sequences:

```
Device(config)#route-map test
Device(config-route-map)# set as-path replace 100
Device(config-route-map)# set as-path replace 6
Device(config-route-map)# set as-path replace 5 78
```

With this configuration, the feature finds all occurrences of 6, 100, and AS sequence "5 78" in the AS-path and replace all of them with own AS. The new AS-Path will be "67 900 65533 900 900 89 900 5 28 900 9".

The following example replaces every AS numbers in the AS-path:

```
Device(config)#route-map test
Device(config-route-map)# set as-path replace any
```

In this configuration, the new AS-Path will be "900 900 900 900 900 900 900 900 900 900 900".

Feature Information for BGP Replace ASNs in the AS Path

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 156: Feature Information for BGP Replace ASNs in the AS Path

Feature Name	Releases	Feature Information
BGP Replace ASNs in the AS Path	Cisco IOS XE Amsterdam 17.1.1	BGP Replace Autonomous System Numbers (ASNs) feature is a BGP policy on the router that allows replacing any of the configured AS number with its own AS. Many AS numbers can be specified to allow such a replace to happen to more than one AS number using the AS_PATH attribute. The following command has been introduced: • **set as-path replace**

CHAPTER 129

Configuring Graceful Insertion and Removal

Graceful Insertion and Removal (GIR) provides an alternative method to minimize network service impact caused by device maintenance. GIR leverages redundant paths in the network to smoothly remove a device under maintenance, out of service, and insert it back to service when the maintenance is complete. This module describes the how to configure GIR.

Restrictions for Graceful Insertion and Removal

GIR is supported on the Cisco ASR 1000 series of routers only for BGP. GIR is configured either by creating customized templates or without a template.

Information About Graceful Insertion and Removal

Overview

Graceful Insertion and Removal (GIR) isolates a router from the network for debugging or an upgrade. The router can be put into maintenance mode using the **start maintenance** command. When router maintenance is complete, the router returns to normal mode on either reaching the configured maintenance timeout, or when the **stop maintenance** command is used.

Create a maintenance mode template before you put the router in maintenance mode. The objective of maintenance mode for a router is to minimize traffic disruption at the time of removal from the network, as well as during the time of insertion. There are mainly three stages:

- Graceful removal of the node from network.
- Performing maintenance on the device.
- Graceful insertion into the network.

A router can be put into maintenance mode using default template or a custom template.

Snapshot Template

Snapshots are taken automatically while entering and exiting the maintenance mode. You can use the **snapshot-template** *template-name* command to capture and store snapshots for pre-selected features. Snapshots are useful to compare the state of a router before it went into maintenance mode and after it came back to normal mode. The snapshot process consists of three parts:

- Creating a snapshot of the states of a few preselected features on the router and storing them on the persistent storage media.
- Listing the snapshots taken at various time intervals and managing them.
- Comparing snapshots and showing the summary and details of each feature.

You can create multiple snapshot templates. But you can apply only one snapshot template at a time. If you do not specify a custom snapshot template, a default snapshot template is applied.

Snapshot templates can be created to generate specific snapshots. A new snapshot template can be created using the **snapshot-template** *template-name* command. The command **snapshot-template** *default-snapshot-template* can be used to specify the default snapshot template in the maintenance mode.

The following example shows the creation of a snapshot template named `gir_1`:

```
snapshot-template gir_1
 router bgp
!
```

Maintenance Template

As a network administrator, you can create a maintenance template that is applied when the system goes into maintenance mode. This allows you to isolate specific protocol instances. All instances that need to be isolated must be explicitly specified.

You can create multiple templates with different configurations. However, only a single template is applied to the maintenance mode CLI. Once applied, the template cannot be updated. If the template has to be updated, then you must remove it, make the changes, and then re-apply.

```
maintenance-template t1
  router bgp 100
```

System Mode Maintenance Counters

GIR has counters to track the following events:

- Number of times the router went into maintenance.
- Ack statistics per client.
- Nack statistics per client
- Number of times a particular client did not acknowledge.
- Number of times switchover happened during GIR. GIR infra will rsync this counter to track multiple switchovers.

- Number of times the failsafe timer expired.
- Number of times system got out of maintenance on a timeout expiry.

Enter the **show system mode maintenance counters** command in privileged EXEC mode, to display the counters that are being tracked by the feature.

Enter the **clear system mode maintenance counters** command in privileged EXEC mode, to clear the counters supported by the feature.

The client-ack timeout value can be configured using the **failsafe** *failsafe-timeout-value* command. Failsafe time is the time that the GIR engine allows a client to transition. Each client sends a notification to the GIR engine about its transition. If it takes more than the failsafe time to transition, it is assumed to have transitioned. The failsafe timer can be configured between 5 - 180 minutes, with a default of 30 minutes.

How to Configure Graceful Insertion and Removal

Creating a Maintenance Template

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **maintenance-template** *template_name*
4. **router** *routing_protocol instance_id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**maintenance-template** *template_name* **Example:** `Device(config)# maintenance-template gir1`	Creates a template with the specified name. For example, see Examples: Creating customer profile.
Step 4	**router** *routing_protocol instance_id* **Example:** `Device(config-maintenance-templ)# router bgp AS-number`	Creates instances that should be isolated under this template. • **router:** Configures routing protocols and associated instance id.

Configuring System Mode Maintenance

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **system mode maintenance**
4. **timeout** *timeout-value* | **template** *template-name* | **snapshot-template** *snapshot name* | **failsafe** *failsafe-timeout-value* | **on-reload reset-reason maintenance**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**system mode maintenance** **Example:** `Device(config)# system mode maintenance`	Enters system mode maintenance configuration mode. Different sub commands to create maintenance mode parameters are configured in this mode.
Step 4	**timeout** *timeout-value* | **template** *template-name* | **snapshot-template** *snapshot name* | **failsafe** *failsafe-timeout-value* | **on-reload reset-reason maintenance**	Configures maintenance mode parameters. • **timeout:** Configures maintenance mode timeout period in minutes, after which the system automatically returns to normal mode. The default timeout value is never. • **template:** Configures maintenance mode using the specified maintenance template. • **snapshot-template:** Configures maintenance mode using the specified snapshot template. • **failsafe:**Configures client-ack timeout value. If the system is going into maintenance mode, it will continue to reach maintenance.If the system is exiting from maintenance mode, then it will reach normal mode. • **on-reload reset-reason maintenance:**Configures the system such that when the system is reloaded it enters the maintenance mode. If it is not configured the system enters the normal mode when it is reloaded.

Starting and Stopping Maintenance Mode

SUMMARY STEPS

1. **enable**
2. **start maintenance**
3. **stop maintenance**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**start maintenance** **Example:** `Device# start maintenance`	Puts the system into maintenance mode.
Step 3	**stop maintenance** **Example:** `Device# stop maintenance`	Puts the system back into normal mode.

Monitoring Graceful Insertion and Removal

Use the following commands to check the status of or display statistics generated by the GIR feature:

Table 157: Privileged EXEC Commands

Command	Purpose
show system mode [maintenance [clients | template *template-name*]]	Displays information about system mode.
show system snapshots [dump *<snapshot-file-name>]*	Displays all the snapshots present on the device.
show system snapshots [dump *<snapshot-file-name>]***xml**	Displays all the snapshots present on the device in XML format.
show system snapshots compare *snapshot-name1 snapshot-name2*	Displays differences between snapshots taken before entering maintenance mode and after exiting from the maintenance mode.

Table 158: Global Configuration Commands for Troubleshooting

Command	Purpose
debug system mode maintenance	Displays information to help troubleshoot the GIR feature.

Configuration Examples for Graceful Removal and Insertion

The following examples show the sequence followed to enable GIR during a maintanence window.

Example: Configuring Snapshot and Maintenance Templates

This example shows how to configure a snapshot template gir_1 with a BGP routing protocol instance.

```
Device#configure terminal
Device(config)#snapshot-template gir_1
Device(config-maintenance-templ)#router BGP 1
```

This example shows how to configure a maintenance template t1 with a BGP routing protocol instance.

```
Device#configure terminal
Device(config)#maintenance-template t1
Device(config-maintenance-templ)#router BGP 1
```

Example: Configuring System Mode Maintenance

This example shows how to create a maintenance template and configure the maintenance mode parameters.

```
Device# configure terminal
Device(config)# system mode maintenance
Device(config-maintenance)# timeout 20
Device(config-maintenance)# failsafe 30
Device(config-maintenance)# on-reload reset-reason maintenance
Device(config-maintenance)# template t1
Device(config-maintenance)# snapshot-template gir_1
Device(config-maintenance)# exit
```

Example: Starting and Stopping the Maintenance Mode

This example shows how to put the system into maintenance mode.

```
Device#start maintenance
Template t1 will be applied. Do you want to continue?[confirm]
Device#
Device(maint-mode)#
```

After the activity is completed, the system can be put out of maintenance mode.

This example shows how to put the system out of maintenance mode.

```
Device(maint-mode)#stop maintenance
Device(maint-mode)#
Device#
```

Example: Displaying System Mode Settings

This example shows how to display system mode settings using different options.

```
Device# show system mode
     System Mode: Normal


Device# show system mode maintenance
     System Mode: Normal
     Current Maintenance Parameters:
     Maintenance Duration: 20(mins)
     Failsafe Timeout: 30(mins)
     Maintenance Template: t1
     Snapshot Template: gir_1
     Reload in Maintenance: True


Device# show system mode maintenance clients
     System Mode: Normal
     Maintenance Clients:
     CLASS-EGP
     router bgp 1000: Transition None

     CLASS-IGP

     CLASS-MCAST

     CLASS-FHRP

     CLASS-L2


Device# show system mode maintenance template default
     System Mode: Normal
     default maintenance-template details:
     router bgp 1000


Device# show system mode maintenance template t1
     System Mode: Normal
     Maintenance Template t1 details:
     router bgp 1000
```

Example: Displaying System Differences Between Entering and Exiting Maintenance Mode

This example shows how to display differences between snapshots taken before entering maintenance mode and after exiting maintenance mode.

```
Device#show system snapshots compare before_maintenance after_maintenance
================================================================================
Feature        Tag                           before_maintenance  after_maintenance
================================================================================
[bgp_summary]

--------------------------------------------------------------------------------
      [Neighbor:192.168.10.2]
               MsgRcvd                       2                   **7**
               up                            00:07:30            **00:11:29**
```

Feature History and Information for Graceful Insertion and Removal

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 159: Feature History and Information for Graceful Insertion and Removal

Feature Name	Release	Feature Information
Graceful Insertion and Removal	Cisco IOS XE Gibraltar 16.12.1	This feature was introduced on the Cisco ASR 1000 series of routers with support for BGP.

CHAPTER 130

BGP Large Community

The BGP large communities attribute provides the capability for tagging routes and modifying BGP routing policy on routers. BGP large communities can be appended or removed selectively on the large communities attribute as the route travels from router to router. The BGP large communities are similar attributes to BGP communities, but with a twelve octet size. The large communities attribute specifies an unordered set of non-duplicated large communities. However, there are no well-known large communities as in communities. The BGP large communities are split logically into a 4 octet Global Administrator field and a 8 octet Local Administrator field. A 4 octet Autonomous System can fit into the Global Administrator field.

Information About the BGP Large Community Feature

BGP Large Community Overview

The BGP large communities attribute provides the capability for tagging routes and modifying BGP routing policy on routers. BGP large communities can be appended or removed selectively on the large communities attribute as the route travels from router to router. When large communities are specified in commands, they are specified as three non-negative decimal integers separated by colons. For example as 1:2:3. The first interger represents the Global Administrator field, and the other two integers represent the Local Administrator field.

The BGP large communities attribute behaves similar to regular communities and is used for the similar purposes. For more information on BGP large community, see the rfc8092 document.

Large Community Lists

A BGP large community list is used to create groups of large communities which can be used in a match clause of a route map. You can use large communities to control the routing policy. Routing policy allows you to filter the routes you receive or advertise, or modify the attributes of the routes you receive or advertise. You can also use a large community list to delete the large communities selectively. There are two types of large community lists:

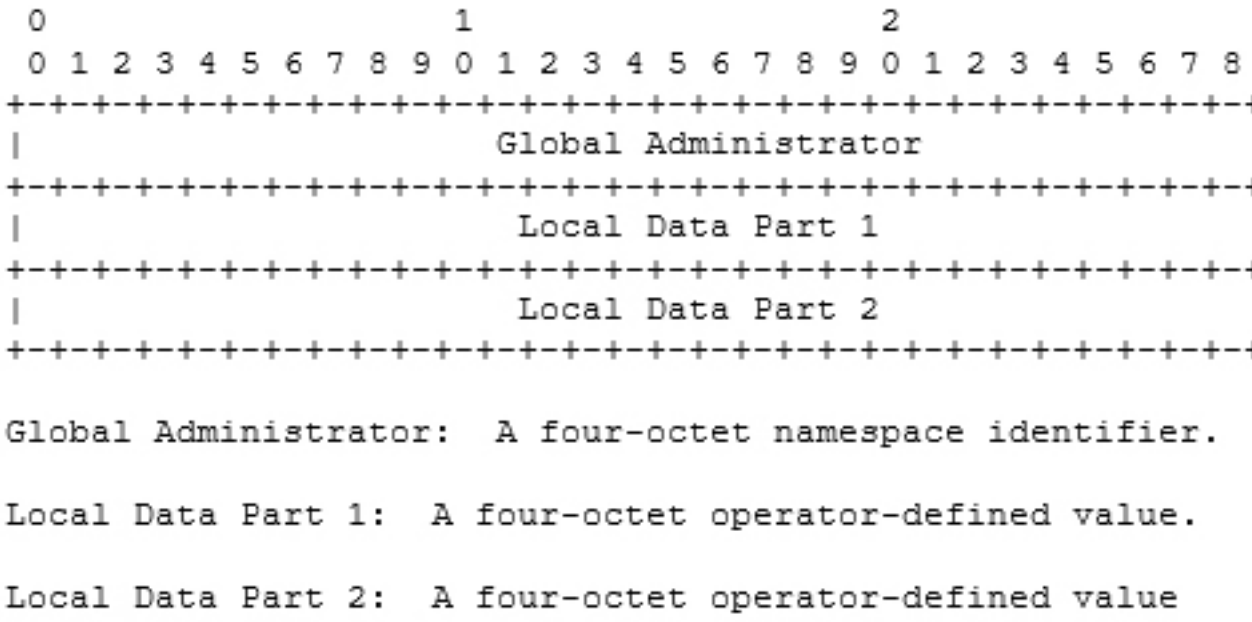
- Standard large community lists-—Specifies large communities.
- Expanded large community lists—Specifies large communities using a regular expression.

A large community list can be either named or numbered. Both named and numbered large community lists can be either standard or expanded. All the rules of numbered large community lists apply to named large community lists, except that there is no limit on the number of named community lists that can be configured.

Note A maximum of 99 (range 1-99) numbered standard large community lists and 401 (range 100-500) numbered expanded large community lists can be configured. Named large community lists do not have this limitation.

BGP Large Communities Attribute

In a BGP large community, the large community value is encoded as a 12 octet number. The following image displays the syntax of the large communities attribute.



How to Configure the BGP Large Community

Enabling BGP Large Communities

To enable large communities, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *IP address* **remote-as** *autonomous-system-number*
4. **address-family** {**ipv4 | ipv6**} {**unicast | multicast**}
5. **neighbor** *IP address* **activate**
6. **neighbor** *IP address* **send-community** [**both | extended | standard**]
7. **exit**
8. **exit**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 64496`	Enters router configuration mode for the specified routing process.
Step 3	**neighbor** *IP address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router)# neighbor 209.165.201.1 remote-as 100`	Enters global address family configuration mode.
Step 4	**address-family** {**ipv4	ipv6**} {**unicast	multicast**} **Example:** `Device(config-router-neighbor)# address-family ipv4 multicast`	Enters global address family configuration mode. **Note** It also supports other available address families.
Step 5	**neighbor** *IP address* **activate** **Example:** `Device(config-router-af)# neighbor 209.165.201.1 activate`	Enters global address family configuration mode and activates the BGP neighbor.
Step 6	**neighbor** *IP address* **send-community** [**both	extended	standard**] **Example:**	Configures the router to send the large communities attribute to the neighbor 209.165.201.1.

	Command or Action	Purpose
	`Device(config-router-neighbor-af)# neighbor 209.165.201.1 send-community standard`	• Both—Sends extended community, large community, and standard communities attributes to the neighbor. • Extended—Sends the extended communities attribute to the neighbor. • Standard—Sends large community and also standard communities attribute to the neighbor. **Note** When configuring the command, not specifying any keyword is equivalent to configuring the standard keyword (no keyword will be displayed in the configuration). When configuring both standard keyword and extended keyword, that will be equivalent to configuring both keyword (both keyword is displayed in the configuration).
Step 7	**exit** **Example:** `Device(config-router-neighbor-af)# exit`	Exits address-family mode and enters global configuration mode.
Step 8	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 9	**end** **Example:** `Device(config)# end`	Exits configuration mode and enters privileged EXEC mode.

Defining a BGP Large Community List

To define a BGP large community list, perform the following steps. BGP large community supports named and numbered community lists.

SUMMARY STEPS

1. **configure terminal**
2. **ip large-community-list** {*list-number* | **standard** *list-name* } {**deny** | **permit**} *community-number large-community*
3. **ip large-community-list** {*list-number* | **expanded** *list-name*} {**deny** | **permit**} *regexp*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 2	**ip large-community-list** {*list-number* | **standard** *list-name* } {**deny | permit**} *community-number large-community*
	Example:
	Numbered Large Community List
	`Device(config)# ip large-community-list 1 permit 1:2:3 5:6:7`
	`Device(config)# ip large-community-list 1 permit 4123456789:4123456780:4123456788`
	Named Large Community List
	`Device(config)# ip large-community-list standard LG_ST permit 1:2:3 5:6:7`
	`Device(config)# ip large-community-list standard LG_ST permit 4123456789:4123456780:4123456788`
		Defines a standard large community list. A standard large community list is composed of a set of entries, each specifying a set of large community lists.

Step 3	**ip large-community-list** {*list-number* | **expanded** *list-name*} {**deny | permit**} *regexp*
	Example:
	Numbered Extended Large Community List
	`Device(config)# ip large-community-list 100 permit ^5:.*:7$`
	`Device(config)# ip large-community-list 100 permit ^5:.*:8$`
	Named Extended Large Community List
	`Device(config)# ip large-community-list expanded LG_EX permit ^5:.*:7$`
	`Device(config)# ip large-community-list expanded LG_EX permit ^5:.*:8$`
		Defines an expanded large community list. An expanded large community list is composed of a set of entries, each specifying a regular expression used to match a set of large communities.

Step 4	**end** **Example:** `Device(config)# end`	Exits configuration mode and enters privileged EXEC mode.

Matching Large Communities

To match BGP large communities, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *map-tag* **[permit | deny]** *[sequence number]*
3. **match large-community** {*list-name* | *list-numbered* }
4. **exit**
5. **route-map** *map-tag* **[permit | deny]** *[sequence number]*
6. **match large-community** {*list-name* | *list-numbered* } **exact-match**
7. **exit**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**route-map** *map-tag* **[permit | deny]** *[sequence number]* **Example:** `Device(config)# route-map test permit 10`	Enters route-map configuration mode.
Step 3	**match large-community** {*list-name* | *list-numbered* } **Example:** `Device(config-route-map)# match large-community 1`	Matches a large community list. Matching a standard large community list entry means that all the large communities defined in such entry are included in the large communities attribute in the route we are trying to match. Matching an expanded large community list entry means the regular expression defined in such entry matches the string representing (in order) all the large communities in the large communities attribute. Matching a large community list means matching at least one of its entries with a grant permit. The entries are evaluated in order. If the first entry in matching has a grant permit, we consider the large community list has matched. If the first entry in matching has a grant deny, or there is no entry matching, we consider the large community list has not matched. **Note** You can specify more than one large community list. In such a case, a match of any large community list will be consider a global match for the match large community statement.
Step 4	**exit** **Example:**	Exits route-map configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-route-map)# exit`	
Step 5	**route-map** *map-tag* **[permit | deny]** *[sequence number]* **Example:** `Device(config)# route-map test permit 20`	Enters the route-map configuration mode and defines the conditions for routes from one routing protocol into another.
Step 6	**match large-community** {*list-name* | *list-numbered* } **exact-match** **Example:** `Device(config-route-map)# match large-community 1 exact-match`	The key word exact-match ensures that there is no large community in the route that is not matched by a large community in the large community list entry. In other words, the set of large communities in the route must be an exact match of the set of large communities in the large community list entry. **Note** The exact-match keyword is only supported for standard large community lists.
Step 7	**exit** **Example:** `Device(config-router-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 8	**end** **Example:** `Device(config)# end`	Exits configuration mode and enters privileged EXEC mode.

Setting BGP Large Communities

To set large communities, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *map-tag* **[permit | deny]** *[sequence number]*
3. **set large-community** { **none | xx1:yy1:zz1....xxn:yyn:zzn**}
4. **exit**
5. **route-map** *map-tag* **[permit | deny]** *[sequence number]*
6. **set large-community xx1:yy1:zz1....xxn:yyn:zzn additive**
7. **exit**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**route-map** *map-tag* **[permit | deny]** *[sequence number]* **Example:** `Device(config)# route-map foo permit 10`	Enters the route-map configuration mode.
Step 3	**set large-community** { **none | xx1:yy1:zz1....xxn:yyn:zzn**} **Example:** `Device(config-route-map)# set large-community 1:2:3 5:6:7`	This route-map set statement is used to set one or more large communities in a route. The keyword **none** sets an empty set of large communities. This is equivalent to an update with no large communities attribute.
Step 4	**exit** **Example:** `Device(config-router-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 5	**route-map** *map-tag* **[permit | deny]** *[sequence number]* **Example:** `Device(config)# route-map foo permit 20`	Enters the route-map configuration mode.
Step 6	**set large-community xx1:yy1:zz1....xxn:yyn:zzn additive** **Example:** `Device(config-route-map)# set large-community 1:2:3 5:6:7 additive`	This route-map set statement is used to set one or more large communities in a route in an additive manner. The keyword **additive** adds the speified large communities without removing the existing large communities.
Step 7	**exit** **Example:** `Device(config-router-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 8	**end** **Example:** `Device(config)# end`	Exits configuration mode and enters privileged EXEC mode.

Deleting Large Communities

To delete BGP large communities, perform the following steps.

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *map-tag* **[permit | deny]** *[sequence number]*
3. **set largecomm-list** {*standard* | *expanded* | *large-community-list number* } **delete**
4. **exit**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**route-map** *map-tag* **[permit | deny]** *[sequence number]* **Example:** `Device(config)# route-map test permit 10`	Enters the route-map configuration mode.
Step 3	**set largecomm-list** {*standard* | *expanded* | *large-community-list number* } **delete** **Example:** `Device(config-route-map)# set largecomm-list 1 delete`	Deletes the large communities based on the matches for the large community list.
Step 4	**exit** **Example:** `Device(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 5	**end** **Example:** `Device(config)# end`	Exits configuration mode and enters privileged EXEC mode.

Verifying the Configuration of the BGP Large Community

To verify the BGP large community, use the following commands.

This example displays entries in the IP version 4 (IPv4) BGP routing table.

```
Device # show bgp ipv4 unicast 2.2.2.2
BGP routing table entry for 2.2.2.2/32, version 2
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
67001
19.0.101.1 from 19.0.101.1 (19.0.101.1)
Origin IGP, localpref 100, valid, external, best
Large Community: 67001:0:2
rx pathid: 0, tx pathid: 0x0
Updated on Nov 1 2020 01:18:02 PST
```

This example shows a list of routes that contain all of the large communities given in the command. The listed routes may contain additional large communities.

```
Device# show bgp large-community 1:2:3 5:6:7
BGP table version is 17, local router ID is 1.1.1.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 5.5.5.5/32       1.1.1.2                  0    100      0 ?
 *>i 5.5.5.6/32       1.1.1.2                  0    100      0 ?
```

This example displays the listed routes that contain only the given large communities when you add the keyword exact-match in the configuration.

```
Device#show bgp large-community 1:2:3 5:6:7 exact-match
BGP table version is 17, local router ID is 1.1.1.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 5.5.5.5/32       1.1.1.2                  0    100      0 ?
```

In the last two examples above, the routes 5.5.5.5/32 and 5.5.5.6/32 contain both the large communities 1:2:3 and 5:6:7. The route 5.5.5.6/32 contains some additional large communities.

This example displays a large community list.

```
Device# show ip largecommunity-list 51
Large Community standard list 51
     permit 1:2:3 5:6:7
```

This example displays a match with large community list.

```
Device# show ip bgp largecommunity-list 51 exact-match
BGP table version is 17, local router ID is 1.1.1.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 5.5.5.5/32       1.1.1.2                  0    100      0 ?
```

Troubleshooting Large Communities

To debug the large communities, use **debug ip bgp update** command.

```
Device# debug ip bgp update

*Mar 10 23:25:01.194: BGP(0): 192.0.0.1 rcvd UPDATE w/ attr: nexthop 192.0.0.1, origin ?,
metric 0, merged path 1, AS_PATH , community 0:44 1:1 2:3, large-community 3:1:244 3:1:245
*Mar 10 23:25:01.194: BGP(0): 192.0.0.1 rcvd 5.5.5.1/32
*Mar 10 23:25:01.194: BGP(0): Revise route installing 1 of 1 routes for 5.5.5.1/32 ->
192.0.0.1(global) to main IP table
```

Memory Display

The **show ip bgp summary** command displays large community memory information.

```
Device # show ip bgp summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 3, main routing table version 3
2 network entries using 496 bytes of memory
2 path entries using 272 bytes of memory
1/1 BGP path/bestpath attribute entries using 288 bytes of memory
1 BGP community entries using 40 bytes of memory
2 BGP large-community entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1096 total bytes of memory
BGP activity 3/1 prefixes, 3/1 paths, scan interval 60 secs
2 networks peaked at 13:04:52 Mar 11 2020 EST (00:07:25.579 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.0.0.2       4            2    1245    1245        3    0    0 18:47:56        0
```

BGP Large Community Configuration Example

The following examples show how to configure policies used to match and manipulate the large communities attribute.

Numbered Standard Large Community List

This example shows how to configure a numbered large community list.

```
ip large-community-list 1 permit 1:2:3 5:6:7
ip large-community-list 1 permit 4123456789:4123456780:4123456788
```

Named Standard Large Community List

This example shows how to configure a named standard large community list.

```
ip large-community-list standard LG_ST permit 1:2:3 5:6:7
ip large-community-list standard LG_ST permit 4123456789:4123456780:4123456788
```

Numbered Expanded Large Community List

This example shows how to configure a numbered expanded large community list.

```
ip large-community-list 100 permit ^5:.*:7$
ip large-community-list 100 permit ^5:.*:8$
```

Named Expanded Large Community List

This example shows how to configure a named expanded large community list.

```
ip large-community-list expanded LG_EX permit ^5:.*:7$
ip large-community-list expanded LG_EX permit ^5:.*:8$
```

Matching Large Communities

These examples show how to match large communities.

```
route-map foo permit 10
 match large-community 1

route-map foo2 permit 10
 match large-community 1 exact-match

route-map foo3 permit 10
 match large-community 100

route-map foo4 permit 10
 match large-community LG_ST exact-match
```

Setting Large Communities

These examples show how to add large communities to the large communities attribute. The *additive* keyword adds the large communities without removing the existing large communities.

```
route-map foo permit 10
 set large-community 1:2:3 5:6:7

route-map foo2 permit 10
 set large-community 1:2:3 5:6:7 additive
```

Deleting Large Communities

These examples show how to remove large communities from the large communities attribute.

```
route-map foo
 set large-comm-list 1 delete

route-map foo2
 set largecomm-list 100 delete

route-map foo3
 set largecomm-list LG_ST delete
```

Additional References

Related Documents

Related Topic	Document Title
BGP commands	Cisco IOS IP Routing: BGP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Standards and RFCs

Standard/RFC	Title
RFC-8092	BGP Large Communities Attribute

Feature Information for BGP Large Communities

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 160: Feature Information for BGP Large Communities

Feature Name	Releases	Feature Information
BGP Large Communities	Cisco IOS XE Bengaluru 17.4.1a	The BGP large communities attribute provides the capability for tagging routes and modifying BGP routing policy on routers.

CHAPTER 131

Support for BGP VPNv4, VPNv6, and EVPN Prefixes with IPv6 Next-Hop

Information About Support for BGP VPNv4, VPNv6, and EVPN Prefixes with IPv6 Next-Hop

Previously, BGP IPv4 or L3VPN standards only had provisions for advertising prefixes with next-hop's AFI/SAFI same as that of prefix's AFI/SAFI in NLRI. The standards were:

- IPv4 prefix NLRI can only include IPv4 next hop
- VPNv4 prefix NLRI can only include RD:IPv4 next hop

From Cisco IOS XE Release 17.8, BGP supports RFC 8950 that allows advertising of VPNv4 prefix with IPv6 next-hop NLRI.

EVPN does not have same restrictions, and it supports advertising of any EVPN prefix with either IPv4 or IPv6 next hop from Cisco IOS XE Release 17.8 onwards.

VPNv4/EVPN prefixes with IPV6 next hops and VPNv6 prefixes with non-IPV4-Mapped-IPv6 next hops are not consumed by the BGP peers. It will be either reflected to an iBGP peer or advertised to an ASBR.

Note Previously, VPNv4 or VPN prefixes were expected to have an MPLS label. If it is IPv6 next hop, even if the MPLS label is not valid, the prefixes are accepted and reflected.

BGP Route Reflector/ASBR Support for IPv6 underlay

Support for VPNv4 with IPv6 next hop is as follows:

- Route-target-filtering is disabled only if it needs to be advertised to an ASBR.
- BGP sends the Extended Next Hop Encoding (ENHE) capability for VPNv4 address-family.
- BGP does not import the VPNv4 prefixes into VRF even if there are matching VRF route-targets.
- BGP advertises the remote VPNv4 prefixes with IPv6 next hop to the RR clients which have sent the EHNE capability.

Support for VPNv6 prefixes with non-IPV4-mapped-IPv6 next hop is as follows:

- Route-target-filtering is disabled only if it needs to be advertised to an ASBR.
- BGP does not import the VPNv6 prefixes into VRF even if there are matching VRF route-targets.
- BGP advertises the remote VPNv6 prefixes with IPv6 next hop to the RR clients.

Support for EVPN with IPv6 next hop is as follows:

- Route-target-filtering is disabled only if it needs to be advertised to an ASBR.
- BGP will not import the EVPN prefixes into VRF/L2RIB even if there are matching stitching VRF route-targets.
- BGP will advertise the EVPN prefixes with V6 next hop to RR clients.

Displaying Information about IPv6 Next Hop

The following table contains the commands that display information related to IPv6 next hop:

Command	Functionality
show bgp vpnv4 unicast all [**detail**] [**prefix**]	Displays the gateway address of VPNv4 prefix
show bgp l2vpn evpn [**route-type**] [**all**] [**prefix**]	BGP EVPN already supports IPv6 gateway for both IPv4 and IPv6 prefixes. This EVPN command displays IPv6 gateway for IPv6 paths.
show ip bgp neighbors	Displays the ENHE capability sent or received.
show bgp ipv4 unicast [**detail**] [**prefix**]	Displays the gateway address of IPv4 prefix.

Example: Displaying BGP Neighbor Connection Parameters

The following example shows BGP connection parameters for the neighbor with the IP address 198.51.100.225.

```
# show bgp vpnv4 unicast all neighbor 198.51.100.225
BGP neighbor is 198.51.100.225, remote AS 1, internal link
BGP version 4, remote router ID 209.165.200.225
BGP state = Established, up for 00:15:24
Last read 00:00:54, last write 00:00:18, hold time is 180, keepalive interval is 60 seconds
Last update received: 00:15:24
Neighbor sessions:
1 active, is not multisession capable (disabled)
Neighbor capabilities:
Route refresh: advertised and received(new)
Four-octets ASN Capability: advertised and received
Address family IPv4 Unicast: advertised and received
Address family VPNv4 Unicast: advertised and received
Address family VPNv6 Unicast: advertised and received
Address family L2VPN Evpn: advertised and received
Enhanced Refresh Capability: advertised and received
Extended Next Hop Encoding Capability:
VPNv4 Unicast: advertised and received
Multisession Capability:
Stateful switchover support enabled: NO for session 1
Message statistics:
InQ depth is 0
OutQ depth is 0
```

In this example output, you can see that VPNv4 Unicast is enabled with the Extended Next Hop Encoding Capability.

Example: Behavior of Route-Map Inbound with Next Hop Set for VPNv4/v6 and EVPN

The following example shows the behavior for route-map inbound set next hop for VPNv4/v6 and EVPN.

```
route-map test
set ip next-hop ..
set ipv6 next-hop ...

router bgp 1
addr vpnv4
neighbor ... route-map test in
```

The **set ipv6 nexthop** rule is ignored. The **set ip next-hop** rule is only applied to VPNv4 prefixes with v4 next hop and *not* applied to VPNv4 prefixes with IPv6 next hop.

```
addr vpnv6
neighbor .... route-map test in
```

The **set ip next-hop** rule is ignored. The **set ipv6 next-hop** rule is only applied to VPNv6 prefixes. The VPNv6 prefixes will only have IPv6 next hops.

```
addr l2vpn evpn
neighbor ... route-map test in
```

The **set ipv6 next-hop** rule is ignored. The **set ip next-hop** rule is only applied to EVPN prefixes with IPv4 next hop and *not* applied to EVPN prefixes with V6 next hop.

Configure Gateway IP

To configure the Gateway IP Address to determine how to reach the specified network prefix, peform these steps:

- Configure Route Map with IPv4 Prefix Lists, on page 1732
- Configure Route Map with IPv6 Prefix Lists, on page 1732
- Set Up a VRF for IPv4 and IPv6 Address Families with an Export Route Map , on page 1733
- Configure BGP and EVPN L2VPN with Gateway IP, on page 1734

Configure Route Map with IPv4 Prefix Lists

To configure route map with IPv4 prefix lists, perform these steps:

Step 1 Create the route map entry and enter the route-map configuration mode.

route-map name {permit | deny}*[sequence-number]*

Example:

```
Device(config)# route-map gateway-v4map permit 10
```

Step 2 Match against one or more IP address prefix lists.

match ipv4 address prefix-list name*[name]*

Example:

```
Device(config-route-map)# match ip address prefix-list gateway-v4list
```

Step 3 Populate the gateway IP with the value from the nexthop.

set evpn gateway-ip use-nexthop

Example:

```
Device(config-route-map)# set evpn gateway-ip use-nexthop
```

Step 4 Create a prefix list to match either IP packets or routes.

ip prefix-list *name* [**seq** *number*] { **permit | deny**} prefix [**eq** *length]* | **ge** *length* | [**le** *length*]]

Example:

```
Device(config)# ip prefix-list gateway-v4list seq 5 permit 100.0.0.0/24
```

Configure Route Map with IPv6 Prefix Lists

To configure route map with IPv6 prefix lists, perform these steps:

Step 1 Create the route map entry and enter the route-map configuration mode.

route-map name {permit | deny} *[sequence-number]*

Example:

```
Device(config)# route-map gateway-v4map permit 10
```

Step 2 Match against one or more IP address prefix lists.
match ipv6 address prefix-list name *[name]*

Example:

```
Device(config-route-map)# match ipv6 address prefix-list gateway-v4list
```

Step 3 Populate the gateway IP with the value from the nexthop.
set evpn gateway-ip use-nexthop

Example:

```
Device(config-route-map)# set evpn gateway-ip use-nexthop
```

Step 4 Create a prefix list to match either IP packets or routes.
ip prefix-list *name* [**seq** *number*] { **permit | deny**} prefix [**eq** *length*] | [**ge** length] | [**le** *length]]*

Example:

```
Device(config)# Device(config)# ipv6 prefix-list gateway-v6list seq 5 permit 100::/64
```

Set Up a VRF for IPv4 and IPv6 Address Families with an Export Route Map

To set up a VRF for IPv4 and IPv6 address families with an export route map, perform these steps:

Before you begin

Step 1 Create a VRF routing table.
vrf definition *vrf-name*

Example:

```
Device(config)# vrf definition red
```

Step 2 Create routing and forwarding tables for the VRF instance.
rd *route-distinguisher*

Example:

```
Device(config-vrf)# rd 1:1
```

Step 3 Configure the IPv4 address family.
address-family { **ipv4 | ipv6**}

Example:

```
Device(config-vrf)# address-family ipv4
```

Step 4 Associate an export map with a VPN Routing and Forwarding (VRF) instance.
export map *map-tag*

Example:

```
Device(config-vrf-af)# export map gateway-v4map
```

Step 5 Exit address family configuration mode.
exit

Example:

```
Device(config-vrf-af)# exit
```

Step 6 Configure the IPv6 address family.
address-family { **ipv4 | ipv6**}

Example:

```
Device(config-vrf)# address-family ipv6
```

Step 7 Associate an export map with a VPN Routing and Forwarding (VRF) instance.
export map *map-tag*

Example:

```
Device(config-vrf-af)# export map gateway-v6map
```

Step 8 Exit address family configuration mode.
exit

Example:

```
Device(config-vrf-af)# exit
```

Configure BGP and EVPN L2VPN with Gateway IP

to configure BGP and EVPN L2VPN with Gateway IP, perform these steps:

Step 1 Configure BGP.
router bgp *number*

Example:

```
Device(config)# router bgp 1000
```

Step 2 Configure the neighbor address to allows BGP sessions to use any operational interface for TCP connections.
neighbor *ip-address* **update-source** *interface-type interface-number*

Example:

```
Device(config-router)# neighbor 2.2.2.2 update-source loopback0
```

Step 3 Configure the neighbor with the remote AS number.
neighbor ip-address remote-as asn

Example:

```
Device(config-router)# neighbor 2.2.2.2 remote-as 2000
```

Step 4 Specify L2VPN address family to enter the address family configuration mode.

address-family l2vpn evpn

Example:

```
Device(config-router)# address-family l2vpn evpn
```

Step 5 Enable the exchange information from a BGP neighbor.

neighbor ip-address activate

Example:

```
Device(config-router-af)# neighbor 2.2.2.2 activate
```

Step 6 Specify the communities attribute sent to a BGP neighbor.

neighbor *ip-address* **send-community extended**

Example:

```
Device(config-router-af)# neighbor 2.2.2.2 send-community extended
```

Step 7 Disable advertisement of gateway IP towards the specified peer.

neighbor ip-address advertise-gw-ip-disable

Example:

```
Device(config-router-af)# neighbor 2.2.2.2 advertise-gw-ip-disable
```

Note Advertise gateway-ip disable option is also accessible through the peer-policy template and peer-group settings.

Step 8 Exit the address family configuration mode.

exit-address-family

Example:

```
Device(config-router-af)# exit-address-family
```

Feature Information for Support for BGP VPNv4, VPNv6, and EVPN Prefixes with IPv6 Next-Hop

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 161: Feature Information for Support for BGP VPNv4, VPNv6, and EVPN Prefixes with IPv6 Next-Hop

Feature Name	Releases	Feature Information
Support for BGP VPNv4, VPNv6, and EVPN Prefixes with IPv6 Next-Hop	17.8.1	This feature allows you to use the Multiprotocol BG (BGP-MP) capability to carry VPNv4 Network Layer Reachability Information (NLRI) in an IPv6 next hop. This helps to reduce the operating cost by carrying both VPNv4 and IPv6 over the same BGP session. VPNv4 or EVPN prefixes with IPv6 next hops and VPNv6 prefixes with non-IPv4-mapped-IPv6 next hops are not supported by the BGP peers. It is either reflected to an iBGP peer or advertised to an ASBR.

PART IV

EIGRP

CHAPTER 132

EIGRP

The Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. The convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP, and IGRP is now obsolete.

The convergence technology of EIGRP is based on an algorithm called the Diffusing Update Algorithm (DUAL). The algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize. Devices that are not affected by topology changes are not involved in recomputations.

Information About Configuring EIGRP

EIGRP Features

- Increased network width--With IP Routing Information Protocol (RIP), the largest possible width of your network is 15 hops. When EIGRP is enabled, the largest possible width is increased to 100 hops, and the EIGRP metric is large enough to support thousands of hops.
- Fast convergence--The DUAL algorithm allows routing information to converge as quickly as any currently available routing protocol.
- Partial updates--EIGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for EIGRP packets.
- Neighbor discovery mechanism--This simple protocol-independent hello mechanism is used to learn about neighboring devices.
- Scaling--EIGRP scales to large networks.

EIGRP Autonomous System Configuration

Configuring the **router eigrp** command with the *autonomous-system-number* argument creates an EIGRP configuration called the EIGRP autonomous system configuration, or EIGRP classic mode. The EIGRP autonomous system configuration creates an EIGRP routing instance that can be used for exchanging routing information.

In EIGRP autonomous system configurations, EIGRP VPNs can be configured only under IPv4 address family configuration mode. A virtual routing and forwarding (VRF) instance and a route distinguisher must be defined before the address family session can be created.

When the address family is configured, we recommend that you configure an autonomous system number either by using the *autonomous-system-number* argument with the **address-family** command or by using the **autonomous-system** command.

EIGRP Named Configuration

Configuring the **router eigrp** command with the *virtual-instance-name* argument creates an EIGRP configuration referred to as the EIGRP named configuration or EIGRP named mode. An EIGRP named configuration does not create an EIGRP routing instance by itself; it is a base configuration that is required to define address-family configurations that are used for routing.

In EIGRP named configurations, EIGRP VPNs can be configured in IPv4 and IPv6 named configurations. A VRF instance and a route distinguisher must be defined before the address family session can be created.

A single EIGRP routing process can support multiple VRFs. The number of VRFs that can be configured is limited only by the available system resources on the device, which is determined by the number running processes and available memory. However, only a single VRF can be supported by each VPN, and redistribution between different VRFs is not supported.

EIGRP Neighbor Relationship Maintenance

Neighbor relationship maintenance is the process that devices use to dynamically learn of other devices on their directly attached networks. Devices must also discover when their neighbors become unreachable or inoperative. Neighbor relationship maintenance is achieved with low overhead by devices when they periodically send small hello packets to each other. As long as hello packets are received, the Cisco software can determine whether a neighbor is alive and functioning. After the status of the neighbor is determined, neighboring devices can exchange routing information.

The reliable transport protocol is responsible for the guaranteed, ordered delivery of Enhanced Interior Gateway Routing Protocol (EIGRP) packets to all neighbors. The reliable transport protocol supports intermixed transmission of multicast and unicast packets. Some EIGRP packets (such as updates) must be sent reliably; this means that the packets require acknowledgment from the destination. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities, hello packets need not be sent reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello packet with an indication in the packet informing receivers that the packet need not be acknowledged. The reliable transport protocol can send multicast packets quickly when unacknowledged packets are pending, thereby ensuring that the convergence time remains low in the presence of varying speed links.

Some EIGRP remote unicast-listen (any neighbor that uses unicast to communicate) and remote multicast-group neighbors may peer with any device that sends a valid hello packet, thus causing security concerns. By authenticating the packets that are exchanged between neighbors, you can ensure that a device accepts packets only from devices that know the preshared authentication key.

Neighbor Authentication

The authentication of packets being sent between neighbors ensures that a device accepts packets only from devices that have the same preshared key. If this authentication is not configured, you can intentionally or accidentally add another device to the network or send packets with different or conflicting route information onto the network, resulting in topology corruption and denial of service (DoS).

Enhanced Interior Gateway Routing Protocol (EIGRP) authentication is configurable on a per-interface basis; packets exchanged between neighbors connected through an interface are authenticated. EIGRP supports message digest algorithm 5 (MD5) authentication to prevent the introduction of unauthorized information from unapproved sources. MD5 authentication is defined in RFC 1321.

DUAL Finite State Machine

The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information (known as the metric) to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A successor is a neighboring device (used for packet forwarding) that has the least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors but only neighbors advertising the destination, a recomputation must occur to determine a new successor. The time required to recompute the route affects the convergence time. Recomputation is processor-intensive, and unnecessary recomputation must be avoided. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, DUAL will use any feasible successors it finds to avoid unnecessary recomputation.

Protocol-Dependent Modules

Protocol-dependent modules are responsible for network-layer protocol-specific tasks. An example is the EIGRP module, which is responsible for sending and receiving EIGRP packets that are encapsulated in the IP. The EIGRP module is also responsible for parsing EIGRP packets and informing DUAL about the new information received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IP routing table. Also, EIGRP is responsible for redistributing routes learned from other IP routing protocols.

Goodbye Message

The goodbye message is a feature designed to improve EIGRP network convergence. The goodbye message is broadcast when an EIGRP routing process is shut down to inform adjacent peers about an impending topology change. This feature allows supporting EIGRP peers to synchronize and recalculate neighbor relationships more efficiently than would occur if the peers discovered the topology change after the hold timer expired.

The following message is displayed by devices that run a supported release when a goodbye message is received:

```
*Apr 26 13:48:42.523: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1   (Ethernet0/0)
is down: Interface Goodbye received
```

A Cisco device that runs a software release that does not support the goodbye message can misinterpret the message as a K-value mismatch and display the following error message:

```
*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor    10.1.1.1 (Ethernet0/0)
is down: K-value mismatch
```

Note The receipt of a goodbye message by a nonsupporting peer does not disrupt normal network operation. The nonsupporting peer terminates the session when the hold timer expires. The sending and receiving devices reconverge normally after the sender reloads.

EIGRP Metric Weights

You can use the **metric weights** command to adjust the default behavior of Enhanced Interior Gateway Routing Protocol (EIGRP) routing and metric computations. EIGRP metric defaults (K values) have been carefully selected to provide optimal performance in most networks.

Note Adjusting EIGRP metric weights can dramatically affect network performance. Because of the complexity of this task, we recommend that you do not change the default K values without guidance from an experienced network designer.

By default, the EIGRP composite cost metric is a 32-bit quantity that is the sum of segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. The formula used to scale and invert the bandwidth value is 10^7/minimum bandwidth in kilobits per second. However, with the EIGRP Wide Metrics feature, the EIGRP composite cost metric is scaled to include 64-bit metric calculations for EIGRP named mode configurations.

For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Gigabit Ethernet (GE), and serial lines running from 9600 bits per second to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

Mismatched K Values

EIGRP K values are the metrics that EIGRP uses to calculate routes. Mismatched K values can prevent neighbor relationships from being established and can negatively impact network convergence. The example given below explains this behavior between two EIGRP peers (Device-A and Device-B).

The following configuration is applied to Device-A. The K values are changed using the **metric weights** command. A value of 2 is entered for the *k1* argument to adjust the bandwidth calculation. A value of 1 is entered for the *k3* argument to adjust the delay calculation.

```
Device(config)# hostname Device-A
Device-A(config)# interface serial 0
Device-A(config-if)# ip address 10.1.1.1 255.255.255.0
Device-A(config-if)# exit
Device-A(config)# router eigrp name1
Device-A(config-router)# address-family ipv4 autonomous-system 4533
Device-A(config-router-af)# network 10.1.1.0 0.0.0.255
Device-A(config-router-af)# metric weights 0 2 0 1 0 0 1
```

The following configuration is applied to Device-B, and the default K values are used. The default K values are 1, 0, 1, 0, 0, and 0.

```
Device(config)# hostname Device-B
Device-B(config)# interface serial 0
Device-B(config-if)# ip address 10.1.1.2 255.255.255.0
Device-B(config-if)# exit
Device-B(config)# router eigrp name1
Device-B(config-router)# address-family ipv4 autonomous-system 4533
Device-B(config-router-af)# network 10.1.1.0 0.0.0.255
Device-B(config-router-af)# metric weights 0 1 0 1 0 0 0
```

The bandwidth calculation is set to 2 on Device-A and set to 1 (by default) on Device-B. This configuration prevents these peers from forming a neighbor relationship.

The following error message is displayed on the console of Device-B because the K values are mismatched:

```
*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1 (Ethernet0/0) is
 down: K-value mismatch
```

The following are two scenarios where the above error message can be displayed:

- Two devices are connected on the same link and configured to establish a neighbor relationship. However, each device is configured with different K values.
- One of two peers has transmitted a “peer-termination” message (a message that is broadcast when an EIGRP routing process is shut down), and the receiving device does not support this message. The receiving device will interpret this message as a K-value mismatch.

Routing Metric Offset Lists

An offset list is a mechanism for increasing incoming and outgoing metrics to routes learned via EIGRP. Optionally, you can limit the offset list with either an access list or an interface.

Note Offset lists are available only in IPv4 configurations. IPv6 configurations do not support offset lists.

EIGRP Cost Metrics

When EIGRP receives dynamic raw radio link characteristics, it computes a composite EIGRP cost metric based on a proprietary formula. To avoid churn in the network as a result of a change in the link characteristics, a tunable dampening mechanism is used.

EIGRP uses metric weights along with a set of vector metrics to compute the composite metric for local RIB installation and route selections. The EIGRP composite cost metric is calculated using the formula:

EIGRP composite cost metric = 256*((K1*Bw) + (K2*Bw)/(256 – Load) + (K3*Delay)*(K5/(Reliability + K4)))

EIGRP uses one or more vector metrics to calculate the composite cost metric. The table below lists EIGRP vector metrics and their descriptions.

Table 162: EIGRP Vector Metrics

Vector Metric	Description
bandwidth	The minimum bandwidth of the route, in kilobits per second. It can be 0 or any positive integer. The bandwidth for the formula is scaled and inverted by the following formula: (10^7/minimum bandwidth (Bw) in kilobits per second)
delay	Route delay, in tens of microseconds.
delay reliability	The likelihood of successful packet transmission, expressed as a number between 0 and 255, where 255 means 100 percent reliability and 0 means no reliability.
load	The effective load of the route, expressed as a number from 0 to 255 (255 is 100 percent loading).
mtu	The minimum maximum transmission unit (MTU) size of the route, in bytes. It can be 0 or any positive integer.

EIGRP monitors metric weights on an interface to allow the tuning of EIGRP metric calculations and indicate the type of service (ToS). The table below lists the K values and their defaults.

Table 163: EIGRP K-Value Defaults

Setting	Default Value
K1	1
K2	0
K3	1
K4	0
K5	0

Most configurations use the delay and bandwidth metrics, with bandwidth taking precedence. The default formula of 256*(Bw + Delay) is the EIGRP metric. The bandwidth for the formula is scaled and inverted by the following formula:

(10^7/minimum Bw in kilobits per second)

Note You can change the weights, but these weights must be the same on all devices.

For example, look at a link whose bandwidth to a particular destination is 128 k and the delay is 84,000 microseconds.

By using a cut-down formula, you can simplify the EIGRP metric calculation to 256*(Bw + Delay), thus resulting in the following value:

Metric = 256*(10^7/128 + 84000/10) = 256*86525 = 22150400

To calculate route delay, divide the delay value by 10 to get the true value in tens of microseconds.

When EIGRP calculates the delay for Mobile Ad Hoc Networks (MANET) and the delay is obtained from a device interface, the delay is always calculated in tens of microseconds. In most cases, when using MANET, you will not use the interface delay, but rather the delay that is advertised by the radio. The delay you will receive from the radio is in microseconds, so you must adjust the cut-down formula as follows:

Metric = $(256*(10^7/128) + (84000*256)/10) = 20000000 + 2150400 = 22150400$

Route Summarization

You can configure EIGRP to perform automatic summarization of subnet routes into network-level routes. For example, you can configure subnet 172.16.1.0 to be advertised as 172.16.0.0 over interfaces that have been configured with subnets of 192.168.7.0. Automatic summarization is performed when two or more **network** router configuration or address family configuration commands are configured for an EIGRP process. This feature is enabled by default.

Route summarization works in conjunction with the **ip summary-address eigrp** command available in interface configuration mode for autonomous system configurations and with the **summary-address** (EIGRP) command for named configurations. You can use these commands to perform additional summarization. If automatic summarization is in effect, there usually is no need to configure network-level summaries using the **ip summary-address eigrp** command.

Summary Aggregate Addresses

You can configure a summary aggregate address for a specified interface. If there are specific routes in the routing table, EIGRP will advertise the summary address of the interface with a metric equal to the minimum metric of the specific routes.

Floating Summary Routes

A floating summary route is created by applying a default route and an administrative distance at the interface level or address family interface level. You can use a floating summary route when configuring the **ip summary-address eigrp** command for autonomous system configurations or the **summary-address** command for named configurations. The following scenarios illustrate the behavior of floating summary routes.

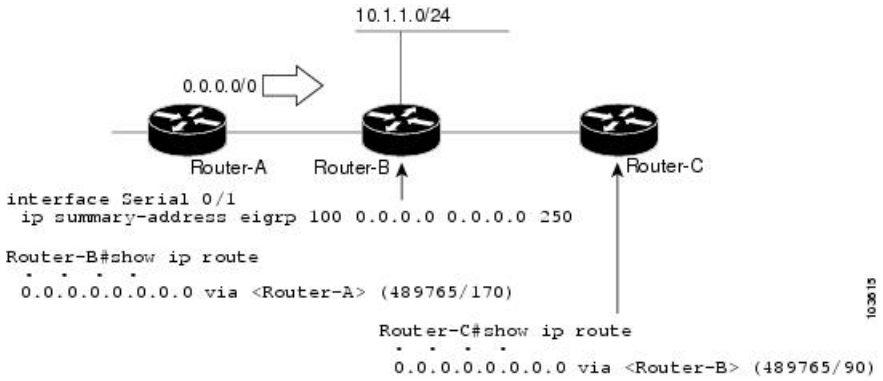
The figure below shows a network with three devices, Device-A, Device-B, and Device-C. Device-A learns a default route from elsewhere in the network and then advertises this route to Device-B. Device-B is configured so that only a default summary route is advertised to Device-C. The default summary route is applied to serial interface 0/1 on Device-B with the following autonomous system configuration:

```
Device-B(config)# interface Serial 0/1
Device-B(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0
```

The default summary route is applied to serial interface 0/1 on Device-B with the following named configuration:

```
Device-B(config)# Router eigrp virtual-name1
Device-B(config-router)# address-family ipv4 unicast vrf vrf1 autonomous-system 1
Device-B(config-router-af)# interface serial 0/1
Device-B(config-router-af-interface)# summary-address 192.168.0.0 255.255.0.0 95
```

Figure 132: Floating Summary Route Applied to Device-B



The configuration of the default summary route on Device-B sends a 0.0.0.0/0 summary route to Device-C and blocks all other routes, including the 10.1.1.0/24 route, from being advertised to Device-C. However, this configuration also generates a local discard route—a route for 0.0.0.0/0 on the null 0 interface with an administrative distance of 5—on Device-B. When this route is created, it overrides the EIGRP-learned default route. Device-B will no longer be able to reach destinations that it would normally reach through the 0.0.0.0/0 route.

This problem is resolved by applying a floating summary route to the interface on Device-B that connects to Device-C. The floating summary route is applied by configuring an administrative distance for the default summary route on the interface of Device-B with the following statement for an autonomous system configuration:

```
Device-B(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0 250
```

The floating summary route is applied by configuring an administrative distance for the default summary route on the interface of Device-B with the following statement for a named configuration:

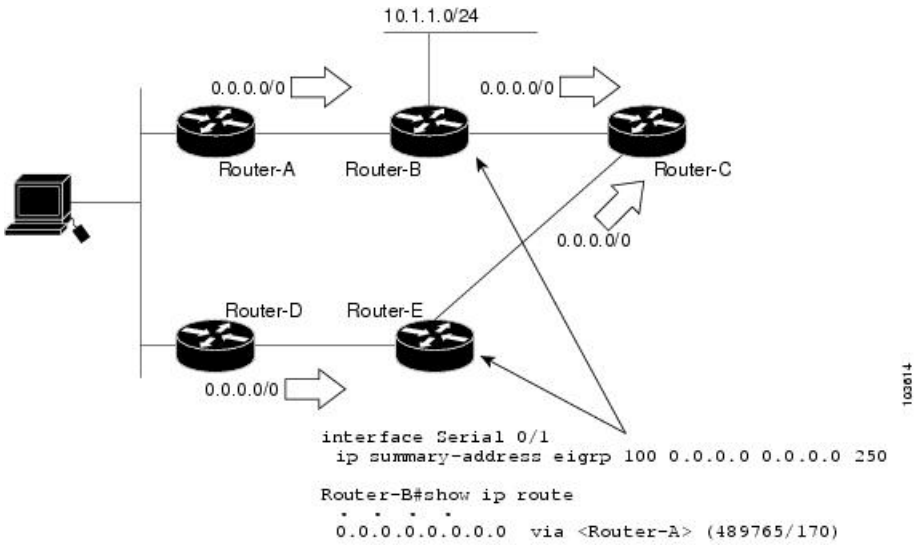
```
Device-B(config)# router eigrp virtual-name1
Device-B(config-router)# address-family ipv4 unicast vrf vrf1 autonomous-system 1
Device-B(config-router-af)# af-interface serial0/1
Device-B(config-router-af-interface)# summary-address eigrp 100 0.0.0.0 0.0.0.0 250
```

The administrative distance of 250, applied in the **summary-address** command, is now assigned to the discard route generated on Device-B. The 0.0.0.0/0, from Device-A, is learned through EIGRP and installed in the local routing table. Routing to Device-C is restored.

If Device-A loses the connection to Device-B, Device-B will continue to advertise a default route to Device-C, which allows traffic to continue to reach destinations attached to Device-B. However, traffic destined to networks connected to Device-A or behind Device-A will be dropped when the traffic reaches Device-B.

The figure below shows a network with two connections from the core, Device-A and Device-D. Both Device-B and Device-E have floating summary routes configured on the interfaces connected to Device-C. If the connection between Device-E and Device-C fails, the network will continue to operate normally. All traffic will flow from Device-C through Device-B to hosts attached to Device-A and Device-D.

Figure 133: Floating Summary Route Applied for Dual-Homed Remotes



However, if the link between Device-A and Device-B fails, the network may incorrectly direct traffic because Device-B will continue to advertise the default route (0.0.0.0/0) to Device-C. In this scenario, Device-C still forwards traffic to Device-B, but Device-B drops the traffic. To avoid this problem, you should configure the summary address with an administrative distance only on single-homed remote devices or areas that have only one exit point between two segments of the network. If two or more exit points exist (from one segment of the network to another), configuring the floating default route can result in the formation of a null route (a route that has quick packet dropping capabilities).

Hello Packets and the Hold-Time Intervals

You can adjust the interval between hello packets and the hold time. Hello packets and hold-time intervals are protocol-independent parameters that work for IP and Internetwork Packet Exchange (IPX).

Routing devices periodically send hello packets to each other to dynamically learn of other devices on their directly attached networks. This information is used to discover neighbors and to learn when neighbors become unreachable or inoperative.

By default, hello packets are sent every 5 seconds. The exception is on low-speed, nonbroadcast multiaccess (NBMA) media, where the default hello interval is 60 seconds. Low speed is considered to be a rate of T1 or slower, as specified with the **bandwidth** interface configuration command. The default hello interval remains 5 seconds for high-speed NBMA networks. Note that for the purposes of EIGRP, Frame Relay and Switched Multimegabit Data Service (SMDS) networks may or may not be considered to be NBMA. These networks are considered NBMA only if the interface has not been configured to use physical multicasting.

You can configure the hold time on a specified interface for a particular EIGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval or 15 seconds. For slow-speed NBMA networks, the default hold time is 180 seconds.

On very congested and large networks, the default hold time might not be sufficient for all devices to receive hello packets from their neighbors. In such cases, you may want to increase the hold time.

Note Do not adjust the hold time without informing your technical support personnel.

Split Horizon

Split horizon controls the sending of EIGRP update and query packets. Split horizon is a protocol-independent parameter that works for IP and IPX. When split horizon is enabled on an interface, update and query packets are not sent to destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a device out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks (such as Frame Relay and SMDS), situations can arise for which this behavior is less than ideal. In such situations and in networks that have EIGRP configured, you may want to disable split horizon.

EIGRP Dual DMVPN Domain Enhancement

The EIGRP Dual DMVPN Domain Enhancement feature supports the **no next-hop self** command on dual Dynamic Multipoint VPN (DMVPN) domains in both IPv4 and IPv6 configurations.

EIGRP, by default, sets the local outbound interface as the next-hop value while advertising a network to a peer, even when advertising routes out of the interface on which the routes were learned. This default setting can be disabled by using the **no ip next-hop-self** command in autonomous system configurations or the **no next-hop-self** command in named configurations. When the **next-hop self** command is disabled, EIGRP does not advertise the local outbound interface as the next hop if the route has been learned from the same interface. Instead, the received next-hop value is used to advertise learned routes. However, this functionality only evaluates the first entry in the EIGRP table. If the first entry shows that the route being advertised is learned on the same interface, then the received next hop is used to advertise the route. The **no next-hop-self** configuration ignores subsequent entries in the table, which may result in the **no-next-hop-self** configuration being dishonored on other interfaces.

The EIGRP Dual DMVPN Domain Enhancement feature introduces the **no-ecmp-mode** keyword, which is an enhancement to the **no next-hop-self** and **no ip next-hop-self** commands. When this keyword is used, all routes to a network in the EIGRP table are evaluated to check whether routes advertised from an interface were learned on the same interface. If a route advertised by an interface was learned on the same interface, the **no next-hop-self** configuration is honored and the received next hop is used to advertise this route.

Link Bandwidth Percentage

By default, EIGRP packets consume a maximum of 50 percent of the link bandwidth when configured with the **bandwidth** interface configuration command for autonomous system configurations and with the **bandwidth-percent** command for named configurations. You might want to change the bandwidth value if a different level of link utilization is required or if the configured bandwidth does not match the actual link bandwidth (which may have been configured to influence route metric calculations). This is a protocol-independent parameter that works for IP and IPX.

EIGRP vNETs

The EIGRP vNET feature uses Layer 3 routing techniques to provide limited fate sharing (the term fate sharing refers to the failure of interconnected systems; that is, different elements of a network are interconnected in such a way that they either fail together or not at all), traffic isolation, and access control with simple configurations. EIGRP virtual network (vNET) configurations are supported in both autonomous-system configurations and named configurations.

The vNET feature allows you to have multiple virtual networks by utilizing a single set of routers and links provided by the physical topology. Routers and links can be broken down into separate virtual networks using separate routing tables and routing processes by using vNETs and VRF configuration commands. The virtual networks facilitate traffic isolation and limited fate sharing. EIGRP's primary role in vNETs is to populate routing tables used by each vNET so that appropriate forwarding can take place. In the vNET model, each vNET effectively has its own complete set of EIGRP processes and resources, thus minimizing the possibility of actions within one vNET affecting another vNET.

The vNET feature supports command inheritance that allows commands entered in interface configuration mode to be inherited by every vNET configured on that interface. These inherited commands, including EIGRP interface commands, can be overridden by vNET-specific configurations in vNET submodes under the interface.

The following are some of the limitations of EIGRP vNETs:

- EIGRP does not support Internetwork Packet Exchange (IPX) within a vNET.
- vNET and VRF configurations are mutually exclusive on an interface. Both VRFs and vNETs can be configured on the router, but they cannot both be defined on the same interface. A VRF cannot be configured within a vNET and a vNET cannot be configured within a VRF.
- Each vNET has its own routing table, and routes cannot be redistributed directly from one vNET into another. EIGRP uses the route replication functionality to meet the requirements of shared services and to copy routes from one vNET Routing Information Base (RIB) to other vNET RIBs.
- Bidirectional Forwarding Detection (BFD) is not supported with EIGRP mode vNET.

EIGRP vNET Interface and Command Inheritance

A vNET router supports two types of interfaces: Edge interface and core (shared) interface.

An edge interface is an ingress point for vNET-unaware networks and is restricted to a single VRF. Use the **vrf forwarding** command to associate the edge interface with a VRF. The **vrf forwarding** command also allows entry into VRF submodes used to define interface settings on a per-VRF basis.

A vNET core interface is used to connect vNET-aware systems and can be shared by multiple vNETs. Use the **vnet trunk** command to enable a core interface.

When the **vnet trunk** command exists on an interface, with or without a VRF list, any EIGRP interface commands on that interface will be applied to the EIGRP instance for every vNET on that interface, including the instance running on the base or the global RIB. If the **vnet trunk** command is deleted from the interface, EIGRP interface commands will remain on and apply to only the global EIGRP instance. If an EIGRP interface command is removed from the main interface, the command will also be removed from every vNET on that interface.

End systems or routing protocol peers reached through an edge interface are unaware of vNETs and do not perform the vNET tagging done in the core of the vNET network.

EIGRP also supports the capability of setting per-vNET interface configurations, which allow you to define interface attributes that influence EIGRP behavior for a single vNET. In the configuration hierarchy, a specific vNET interface setting has precedence over settings applied to the entire interface and inherited by each vNET configured on that interface.

EIGRP provides interface commands to modify the EIGRP-specific attributes of an interface, and these interface commands can be entered directly on the interface for EIGRP autonomous system configurations, or in address family interface configuration mode for the EIGRP named mode configurations.

How to Configure EIGRP

Enabling EIGRP Autonomous System Configuration

Perform this task to enable EIGRP and create an EIGRP routing process. EIGRP sends updates to interfaces in specified networks. If you do not specify the network of an interface, the interface will not be advertised in any EIGRP update.

Configuring the **router eigrp** *autonomous-system-number* command creates an EIGRP autonomous system configuration that creates an EIGRP routing instance, which can be used for tagging routing information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **network** *network-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 1`	Configures an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.

	Command or Action	Purpose
Step 4	**network** *network-number* **Example:** `Device(config-router)# network 172.16.0.0`	Associates a network with an EIGRP routing process.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

EIGRP Named Configuration

Configuring the **router eigrp** command with the *virtual-instance-name* argument creates an EIGRP configuration referred to as the EIGRP named configuration or EIGRP named mode. An EIGRP named configuration does not create an EIGRP routing instance by itself; it is a base configuration that is required to define address-family configurations that are used for routing.

In EIGRP named configurations, EIGRP VPNs can be configured in IPv4 and IPv6 named configurations. A VRF instance and a route distinguisher must be defined before the address family session can be created.

A single EIGRP routing process can support multiple VRFs. The number of VRFs that can be configured is limited only by the available system resources on the device, which is determined by the number running processes and available memory. However, only a single VRF can be supported by each VPN, and redistribution between different VRFs is not supported.

Configuring Optional EIGRP Parameters in an Autonomous System Configuration

Perform this task to configure optional EIGRP parameters, which include applying offsets to routing metrics, adjusting EIGRP metrics, and disabling automatic summarization in an EIGRP autonomous system configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **network** *ip-address* [*wildcard-mask*]
5. **passive-interface** [**default**] [*interface-type interface-number*]
6. **offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*]
7. **metric weights** *tos k1 k2 k3 k4 k5*
8. **no auto-summary**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 1`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 5	**passive-interface** [**default**] [*interface-type interface-number*] **Example:** `Device(config-router)# passive-interface`	(Optional) Suppresses EIGRP hello packets and routing updates on interfaces while still including the interface addresses in the topology database.
Step 6	**offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*] **Example:** `Device(config-router)# offset-list 21 in 10 gigabitethernet 0/0/1`	(Optional) Applies an offset to routing metrics.
Step 7	**metric weights** *tos k1 k2 k3 k4 k5* **Example:** `Device(config-router)# metric weights 0 2 0 2 0 0`	(Optional) Adjusts the EIGRP metric or K value. • EIGRP uses the following formula to determine the total metric to the network: EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4))) **Note** If K5 is 0, then (K5/ (Reliability + K4)) is defined as 1.
Step 8	**no auto-summary** **Example:**	(Optional) Disables automatic summarization. **Note** Automatic summarization is enabled by default.

	Command or Action	Purpose
	`Device(config-router)# no auto-summary`	
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Optional EIGRP Parameters in a Named Configuration

Perform this task to configure optional EIGRP named configuration parameters, which includes applying offsets to routing metrics, adjusting EIGRP metrics, setting the RIB-scaling factor, and disabling automatic summarization.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**unicast**] [**vrf** *vrf-name*] [**multicast**] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **metric weights** *tos k1 k2 k3 k4 k5 k6*
7. **af-interface** *interface-type interface-number*}
8. **passive-interface**
9. **bandwidth-percent** *maximum-bandwidth-percentage*
10. **exit-af-interface**
11. **topology** {**base** | *topology-name* **tid** *number*}
12. **offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*]
13. **no auto-summary**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**unicast**] [**vrf** *vrf-name*] [**multicast**] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 6	**metric weights** *tos k1 k2 k3 k4 k5 k6* **Example:** `Device(config-router-af)# metric weights 0 2 0 2 0 0 0`	(Optional) Adjusts the EIGRP metric or K value. • EIGRP uses the following formula to determine the total 32-bit metric to the network: EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4))) • EIGRP uses the following formula to determine the total 64-bit metric to the network: EIGRP Metric = 256*((K1*Throughput) + (K2*Throughput)/(256-Load) + (K3*Latency)+ (K6*Extended Attributes))*(K5/(Reliability + K4))) **Note** If K5 is 0, then (K5/ (Reliability + K4)) is defined as 1.
Step 7	**af-interface** *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 8	**passive-interface** **Example:** `Device(config-router-af-interface)# passive-interface`	Suppresses EIGRP hello packets and routing updates on interfaces while still including the interface addresses in the topology database.

	Command or Action	Purpose
Step 9	**bandwidth-percent** *maximum-bandwidth-percentage* **Example:** `Device(config-router-af-interface)# bandwidth-percent 75`	Configures the percentage of bandwidth that may be used by an EIGRP address family on an interface.
Step 10	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 11	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 12	**offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*] **Example:** `Device(config-router-af-topology)# offset-list 21 in 10 gigabitethernet 6/2`	(Optional) Applies an offset to routing metrics.
Step 13	**no auto-summary** **Example:** `Device(config-router-af-topology)# no auto-summary`	(Optional) Disables automatic summarization. **Note** Automatic summarization is enabled by default.
Step 14	**end** **Example:** `Device(config-router-af-topology)# end`	Returns to privileged EXEC mode.

Configuring the EIGRP Redistribution Autonomous System Configuration

Perform this task to configure redistribution of non-EIGRP protocol metrics into EIGRP metrics and to configure the EIGRP administrative distance in an EIGRP autonomous system configuration.

You must use a default metric to redistribute a protocol into EIGRP, unless you use the **redistribute** command.

Note Metric defaults have been carefully set to work for a wide variety of networks. Take great care when changing these values.

Default metrics are supported only when you are redistributing from EIGRP or static routes.

An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is an integer from 0 to 255. In general, the higher the value the lower the trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **network** *ip-address* [*wildcard-mask*]
5. **redistribute** *protocol*
6. **distance eigrp** *internal-distance external-distance*
7. **default-metric** *bandwidth delay reliability loading mtu*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 1`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 5	**redistribute** *protocol* **Example:** `Device(config-router)# redistribute rip`	Redistributes routes from one routing domain into another routing domain.
Step 6	**distance eigrp** *internal-distance external-distance* **Example:** `Device(config-router)# distance eigrp 80 130`	Allows the use of two administrative distances—internal and external.
Step 7	**default-metric** *bandwidth delay reliability loading mtu* **Example:**	Sets metrics for EIGRP.

	Command or Action	Purpose
	`Device(config-router)# default-metric 1000 100 250 100 1500`	
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Route Summarization Autonomous System Configuration

Perform this task to configure EIGRP to perform automatic summarization of subnet routes into network-level routes in an EIGRP autonomous system configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **no auto-summary**
5. **exit**
6. **interface** *type number*
7. **no switchport**
8. **bandwidth** *kpbs*
9. **ip summary-address eigrp** *as-number ip-address mask* [*admin-distance*] [**leak-map** *name*]
10. **ip bandwidth-percent eigrp** *as-number percent*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.

	Command or Action	Purpose
Step 4	**no auto-summary** **Example:** `Device(config-router)# no auto-summary`	Disables automatic summarization of subnet routes into network-level routes
Step 5	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0/3`	Enters interface configuration mode.
Step 7	**no switchport** **Example:** `Device(config-if)# no switchport`	Puts an interface into Layer 3 mode
Step 8	**bandwidth** *kpbs* **Example:** `bandwidth 56`	Sets the inherited and received bandwidth values for an interface
Step 9	**ip summary-address eigrp** *as-number ip-address mask* [*admin-distance*] [**leak-map** *name*] **Example:** `Device(config-if)# ip summary-address eigrp 100 10.0.0.0 0.0.0.0`	(Optional) Configures a summary aggregate address.
Step 10	**ip bandwidth-percent eigrp** *as-number percent* **Example:** `Device(config-if)# ip bandwidth-percent eigrp 209 75`	(Optional) Configures the percentage of bandwidth that may be used by EIGRP on an interface.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Route Summarization Named Configuration

Perform this task to configure EIGRP to perform automatic summarization of subnet routes into network-level routes in an EIGRP named configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **af-interface {default** | *interface-type interface-number***}**
6. **summary-address** *ip-address mask* [*administrative-distance* [**leak-map** *leak-map-name*]]
7. **exit-af-interface**
8. **topology** {**base** | *topology-name* **tid** *number*}
9. **summary-metric** *network-address subnet-mask bandwidth delay reliability load mtu*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

	Command or Action	Purpose
Step 5	**af-interface {default | *interface-type interface-number*}** **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**summary-address** *ip-address mask* [*administrative-distance* [**leak-map** *leak-map-name*]] **Example:** `Device(config-router-af-interface)# summary-address 192.168.0.0 255.255.0.0`	Configures a summary address for EIGRP.
Step 7	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 8	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 9	**summary-metric** *network-address subnet-mask bandwidth delay reliability load mtu* **Example:** `Device(config-router-af-topology)# summary-metric 192.168.0.0/16 10000 10 255 1 1500`	(Optional) Configures a fixed metric for an EIGRP summary aggregate address.
Step 10	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Event Logging Autonomous System Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **eigrp event-log-size** *size*
5. **eigrp log-neighbor-changes**
6. **eigrp log-neighbor-warnings** [*seconds*]

7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**eigrp event-log-size** *size* **Example:** `Device(config-router)# eigrp event-log-size 5000010`	(Optional) Sets the size of the EIGRP event log.
Step 5	**eigrp log-neighbor-changes** **Example:** `Device(config-router)# eigrp log-neighbor-changes`	(Optional) Enables logging of EIGRP neighbor adjacency changes. • By default, the system logs EIGRP neighbor adjacency changes to help you monitor the stability of the routing system and detect problems.
Step 6	**eigrp log-neighbor-warnings** [*seconds*] **Example:** `Device(config-router)# eigrp log-neighbor-warnings 300`	(Optional) Enables the logging of EIGRP neighbor warning messages.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Event Logging Named Configuration

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **eigrp log-neighbor-warnings** [*seconds*]
6. **eigrp log-neighbor-changes**
7. **topology** {**base** | *topology-name* **tid** *number*}
8. **eigrp event-log-size** *size*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**eigrp log-neighbor-warnings** [*seconds*] **Example:**	(Optional) Enables the logging of EIGRP neighbor warning messages.

	Command or Action	Purpose
	`Device(config-router-af)# eigrp log-neighbor-warnings 300`	
Step 6	**eigrp log-neighbor-changes** **Example:** `Device(config-router-af)# eigrp log-neighbor-changes`	(Optional) Enables logging of EIGRP neighbor adjacency changes. • By default, the system logs EIGRP neighbor adjacency changes to help you monitor the stability of the routing system and detect problems.
Step 7	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 8	**eigrp event-log-size** *size* **Example:** `Device(config-router-af-topology)# eigrp event-log-size 10000`	(Optional) Sets the size of the EIGRP event log.
Step 9	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring Equal and Unequal Cost Load Balancing Autonomous System Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **traffic-share balanced**
5. **maximum-paths** *number-of-paths*
6. **variance** *multiplier*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**traffic-share balanced** **Example:** `Device(config-router)# traffic-share balanced`	Controls how traffic is distributed among routes when multiple routes for the same destination network have different costs.
Step 5	**maximum-paths** *number-of-paths* **Example:** `Device(config-router)# maximum-paths 5`	Controls the maximum number of parallel routes that an IP routing protocol can support.
Step 6	**variance** *multiplier* **Example:** `Device(config-router)# variance 1`	Controls load balancing in an internetwork based on EIGRP.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Equal and Unequal Cost Load Balancing Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **topology** {**base** | *topology-name* **tid** *number*}

6. **traffic-share balanced**
7. **maximum-paths** *number-of-paths*
8. **variance** *multiplier*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 6	**traffic-share balanced** **Example:** `Device(config-router-af-topology)# traffic-share balanced`	Controls how traffic is distributed among routes when multiple routes for the same destination network have different costs.

	Command or Action	Purpose
Step 7	**maximum-paths** *number-of-paths* **Example:** `Device(config-router-af-topology)# maximum-paths 5`	Controls the maximum number of parallel routes that an IP routing protocol can support.
Step 8	**variance** *multiplier* **Example:** `Device(config-router-af-topology)# variance 1`	Controls load balancing in an internetwork based on EIGRP.
Step 9	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Adjusting the Interval Between Hello Packets and the Hold Time in an Autonomous System Configuration

Note Cisco recommends not to adjust the hold time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **exit**
5. **interface** *type number*
6. **no switchport**
7. **ip hello-interval eigrp** *autonomous-system-number seconds*
8. **ip hold-time eigrp** *autonomous-system-number seconds*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**exit** **Example:** `Device(config-router)# exit`	Exits to global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0/9`	Enters interface configuration mode.
Step 6	**no switchport** **Example:** `Device(config-if)# no switchport`	Puts an interface into Layer 3 mode
Step 7	**ip hello-interval eigrp** *autonomous-system-number seconds* **Example:** `Device(config-if)# ip hello-interval eigrp 109 10`	Configures the hello interval for an EIGRP routing process.
Step 8	**ip hold-time eigrp** *autonomous-system-number seconds* **Example:** `Device(config-if)# ip hold-time eigrp 109 40`	Configures the hold time for an EIGRP routing process. **Note** Do not adjust the hold time without consulting your technical support personnel.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Adjusting the Interval Between Hello Packets and the Hold Time in a Named Configuration

Note Do not adjust the hold time without consulting your technical support personnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **af-interface** {**default** | *interface-type interface-number*}
6. **hello-interval** *seconds*
7. **hold-time** *seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

	Command or Action	Purpose

	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**hello-interval** *seconds* **Example:** `Device(config-router-af-interface)# hello-interval 10`	Configures the hello interval for an EIGRP address family named configuration.
Step 7	**hold-time** *seconds* **Example:** `Device(config-router-af-interface)# hold-time 50`	Configures the hold time for an EIGRP address family named configuration.
Step 8	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.

Disabling the Split Horizon Autonomous System Configuration

Split horizon controls the sending of EIGRP updates and query packets. When split horizon is enabled on an interface, updates and query packets are not sent for destinations for which this interface is the next hop. Controlling updates and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip split-horizon eigrp** *autonomous-system-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**no ip split-horizon eigrp** *autonomous-system-number* **Example:** `Device(config-if)# no ip split-horizon eigrp 101`	Disables split horizon.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling the Split Horizon and Next-Hop-Self Named Configuration

EIGRP, by default, sets the next-hop value to the local outbound interface address for routes that it is advertising, even when advertising those routes back from the same interface from where they were learned. Perform this task to change this default setting and configure EIGRP to use the received next-hop value when advertising these routes. Disabling next-hop-self is primarily useful in DMVPN spoke-to-spoke topologies.

By default, split horizon is enabled on all interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **af-interface** {**default** | *interface-type interface-number*}

6. **no split-horizon**
7. **no next-hop-self [no-ecmp-mode]**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**no split-horizon** **Example:** `Device(config-router-af-interface)# no split-horizon`	Disables EIGRP split horizon.

	Command or Action	Purpose
Step 7	**no next-hop-self [no-ecmp-mode]** **Example:** `Device(config-router-af-interface)# no next-hop-self no-ecmp-mode`	(Optional) Instructs an EIGRP router to use the received next hop rather than the local outbound interface address as the next hop. • The **no-ecmp-mode** keyword is an enhancement to the **no next-hop-self** command. When this optional keyword is enabled, all paths to a network in the EIGRP table are evaluated to check whether routes advertised from an interface were learned on the same interface.
Step 8	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining the EIGRP Autonomous System Configuration

This task is optional. Use the commands in any order desired to monitor and maintain EIGRP autonomous system configuration.

SUMMARY STEPS

1. **enable**
2. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **accounting**
3. **show ip eigrp events** [*starting-event-number ending-event-number*] [**type**]
4. **show ip eigrp interfaces** [**vrf** {*vrf-name*| *****}] [*autonomous-system-number*] [*type number*] [**detail**]
5. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]
6. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]
7. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **traffic**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device# enable
```

Step 2 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **accounting**

Displays prefix accounting information for EIGRP processes.

Example:

```
Device# show ip eigrp vrf VRF1 accounting
```

Step 3 **show ip eigrp events** [*starting-event-number ending-event-number*] [**type**]

Displays information about interfaces that are configured for EIGRP.

Example:

```
Device# show ip eigrp events
```

Step 4 **show ip eigrp interfaces** [**vrf** {*vrf-name*| *****}] [*autonomous-system-number*] [*type number*] [**detail**]

Displays neighbors discovered by EIGRP.

Example:

```
Device# show ip eigrp interfaces
```

Step 5 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]

Displays neighbors discovered by EIGRP

Example:

```
Device# show ip eigrp neighbors
```

Step 6 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]

Displays entries in the EIGRP topology table.

Example:

```
Device# show ip eigrp topology
```

Step 7 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **traffic**

Displays the number of EIGRP packets sent and received.

Example:

```
Device# show ip eigrp traffic
```

Monitoring and Maintaining the EIGRP Named Configuration

This task is optional. Use the commands in any order desired to monitor and maintain the EIGRP named configuration.

SUMMARY STEPS

1. **enable**
2. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting**
3. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **events** [*starting-event-number ending-event-number*] [**errmsg** [*starting-event-number ending-event-number*]] [**sia** [*starting-event-number ending-event-number*]] [**type**]

4. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **interfaces** [**detail**] [*interface-type interface-number*]
5. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **neighbors** [**static**] [**detail**] [*interface-type interface-number*]
6. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **timers**
7. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **topology** [*topology-name*] [*ip-address*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**route-type** {**connected** | **external** | **internal** | **local** | **redistributed** | **summary** | **vpn**}]
8. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **traffic**
9. **show eigrp plugins** [*plugin-name*] [**detailed**]
10. **show eigrp protocols** [**vrf** *vrf-name*]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device# enable
```

Step 2 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting**

Displays prefix accounting information for EIGRP processes.

Example:

```
Device# show eigrp address-family ipv4 22 accounting
```

Step 3 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **events** [*starting-event-number ending-event-number*] [**errmsg** [*starting-event-number ending-event-number*]] [**sia** [*starting-event-number ending-event-number*]] [**type**]

Displays information about EIGRP address-family events.

Example:

```
Device# show eigrp address-family ipv4 3 events
```

Step 4 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **interfaces** [**detail**] [*interface-type interface-number*]

Displays information about interfaces that are configured for EIGRP.

Example:

```
Device# show eigrp address-family ipv4 4453 interfaces
```

Step 5 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **neighbors** [**static**] [**detail**] [*interface-type interface-number*]

Displays the neighbors that are discovered by EIGRP.

Example:

```
Device# show eigrp address-family ipv4 4453 neighbors
```

Step 6 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **timers**

Displays information about EIGRP timers and expiration times.

Example:

```
Device# show eigrp address-family ipv4 4453 timers
```

Step 7 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **topology** [*topology-name*] [*ip-address*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**route-type** {**connected** | **external** | **internal** | **local** | **redistributed** | **summary** | **vpn**}]

Displays entries in the EIGRP topology table.

Example:

```
Device# show eigrp address-family ipv4 4453 topology
```

Step 8 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **traffic**

Displays the number of EIGRP packets that are sent and received.

Example:

```
Device# show eigrp address-family ipv4 4453 traffic
```

Step 9 **show eigrp plugins** [*plugin-name*] [**detailed**]

Displays general information, including the versions of the EIGRP protocol features that are currently running on the device.

Example:

```
Device# show eigrp plugins
```

Step 10 **show eigrp protocols** [**vrf** *vrf-name*]

Displays further information about EIGRP protocols that are currently running on a device.

Example:

```
Device# show eigrp protocols
```

Configuration Examples for EIGRP

Example: Enabling EIGRP—Autonomous System Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
```

Example: Enabling EIGRP—Named Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
```

Example: EIGRP Parameters—Autonomous System Configuration

The following example shows how to configure optional EIGRP autonomous system configuration parameters, including applying offsets to routing metrics, adjusting EIGRP metrics, and disabling automatic summarization:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# passive-interface
Device(config-router)# offset-list 21 in 10 ethernet 0
Device(config-router)# metric weights 0 2 0 2 0 0
Device(config-router)# no auto-summary
Device(config-router)# exit
```

Example: EIGRP Parameters—Named Configuration

The following example shows how to configure optional EIGRP named configuration parameters, including applying offsets to routing metrics, adjusting EIGRP metrics, setting RIB-scaling factor, and disabling automatic summarization.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
Device(config-router-af)# metric weights 0 2 0 2 0 0 0
Device(config-router-af)# metric rib-scale 100
Device(config-router-af)# af-interface gigabitethernet 0/0/1
Device(config-router-af-interface)# passive-interface
Device(config-router-af-interface)# bandwidth-percent 75
Device(config-router-af-interface)# exit-af-interface
```

```
Device(config-router-af-interface)# topology base
Device(config-router-af-topology)# offset-list 21 in 10 gigabitethernet 0/0/1
Device(config-router-af-topology)# no auto-summary
Device(config-router-af-topology)# exit-af-topology
```

Example: EIGRP Redistribution—Autonomous System Configuration

The following example shows how to configure redistribution of non-EIGRP protocol metrics into EIGRP metrics and configure the EIGRP administrative distance in an EIGRP autonomous system configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# redistribute rip
Device(config-router)# distance eigrp 80 130
Device(config-router)# default-metric 1000 100 250 100 1500
```

Example: EIGRP Route Summarization—Autonomous System Configuration

The following example shows how to configure route summarization on an interface and configure the automatic summary feature for an EIGRP autonomous system configuration. The following configuration causes EIGRP to summarize the network from Ethernet interface 0/0.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 101
Device(config-router)# no auto-summary
Device(config-router)# exit
Device(config)# interface Gigabitethernet 1/0/1
Device(config-if)# no switchport
bandwidth 56
Device(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0
Device(config-if)# ip bandwidth-percent eigrp 209 75
```

Note You should not use the **ip summary-address eigrp** summarization command to generate the default route (0.0.0.0) from an interface because this creates an EIGRP summary default route to the null 0 interface with an administrative distance of 5. The low administrative distance of this default route can cause this route to displace default routes learned from other neighbors through the routing table. If the default route learned from the neighbors is displaced by the summary default route, or if the summary route is the only default route present, all traffic destined for the default route will not leave the router; instead, traffic will be sent to the null 0 interface, where it is dropped. The recommended way to send only the default route out of a given interface is to use the **distribute-list** command. You can configure this command to filter all outbound route advertisements sent out from the interface with the exception of the default (0.0.0.0).

Example: EIGRP Route Summarization—Named Configuration

The following example shows how to configure route summarization on an interface and configure the automatic summary feature for an EIGRP named configuration. This configuration causes EIGRP to summarize network 192.168.0.0 only from Ethernet interface 0/0.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# summary-address 192.168.0.0 255.255.0.0
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af)# topology base
Device(config-router-af-topology)# summary-metric 192.168.0.0/16 10000 10 255 1 1500
```

Example: EIGRP Event Logging—Autonomous System Configuration

The following example shows how to configure EIGRP event logging parameters, including setting the size of the EIGRP event log for an EIGRP autonomous system configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# eigrp event-log-size 5000
Device(config-router)# eigrp log-neighbor-changes
Device(config-router)# eigrp log-neighbor-warnings 300
```

Example: EIGRP Event Logging—Named Configuration

The following example shows how to configure EIGRP event logging parameters, including setting the size of the EIGRP event log for an EIGRP named configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# eigrp log-neighbor-warnings 300
Device(config-router-af)# eigrp log-neighbor-changes
Device(config-router-af)# topology base
Device(config-router-af-topology)# eigrp event-log-size 10000
```

Example: Equal and Unequal Cost Load Balancing—Autonomous System Configuration

The following example shows how to configure traffic distribution among routes, the maximum number of parallel routes, and load balancing in an EIGRP named configuration network:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# traffic-share balanced
Device(config-router)# maximum-paths 5
Device(config-router)# variance 1
```

Example: Equal and Unequal Cost Load Balancing—Named Configuration

The following example shows how to configure traffic distribution among routes, the maximum number of parallel routes, and load balancing in an EIGRP named configuration network:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# topology base
Device(config-router-af-topology)# traffic-share balanced
Device(config-router-af-topology)# maximum-paths 5
Device(config-router-af-topology)# variance 1
```

Example: Adjusting the Interval Between Hello Packets and the Hold Time—Autonomous System Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface Gibabitethernet 1/0/9
Device(config-if)# no switchport
Device(config-if)# ip hello-interval eigrp 109 10
Device(config-if)# ip hold-time eigrp 109 40
```

Example: Adjusting the Interval Between Hello Packets and the Hold Time—Named Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# hello-interval 10
Device(config-router-af-interface)# hold-time 50
```

Example: Disabling the Split Horizon—Autonomous System Configuration

Split horizon is enabled on all interfaces by default. The following example shows how to disable split horizon for an EIGRP autonomous system configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface Ethernet 0/1
Device(config-if)# no ip split-horizon eigrp 101
```

Example: Disabling the Split Horizon and Next-Hop-Self—Named Configuration

Split horizon is enabled on all interfaces by default. The following example shows how to disable split horizon in an EIGRP named configuration.

EIGRP, by default, sets the next-hop value to the local outbound interface address for routes that it advertises, even when advertising those routes back out of the same interface from where they were learned. The following example shows how to change this default to instruct EIGRP to use the received next-hop value when advertising these routes in an EIGRP named configuration. Disabling the **next-hop-self** command is primarily useful in DMVPN spoke-to-spoke topologies.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# no split-horizon
Device(config-router-af-interface)# no next-hop-self no-ecmp-mode
```

Example: Command Inheritance and Virtual Network Interface Mode Override in an EIGRP Environment

Suppose a GigabitEthernet interface is configured with the following EIGRP commands:

```
interface gigabitethernet 0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 end
```

Because a trunk is configured, a VRF subinterface is automatically created and the commands on the main interface are inherited by the VRF subinterface (g0/0/0.3, where the number 3 is the tag number from vnet tag 3.)

Use the **show derived-config** command to display the hidden subinterface. The following sample output shows that all the commands entered on GigabitEthernet 0/0/0 have been inherited by GigabitEthernet 0/0/0.3:

```
Device# show derived-config interface gigabitethernet 0/0/0.3

Building configuration...
Derived configuration : 478 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET vrf1
 vrf forwarding vrf1
 encapsulation dot1Q 3
 ip address 192.0.2.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
```

```
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 end
```

Use the virtual network interface mode to override the commands entered in interface configuration mode. For example:

```
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# vnet name vrf1
Device(config-if-vnet)# no ip authentication mode eigrp 1 md5
! disable authen for e0/0.3 only
Device(config-if-vnet)# ip authentication key-chain eigrp 1 y
! different key-chain
Device(config-if-vnet)# ip band eigrp 1 99
! higher bandwidth-percent
Device(config-if-vnet)# no ip dampening-change eigrp 1
! disable dampening-change
Device(config-if-vnet)# ip hello eigrp 1 7
Device(config-if-vnet)# ip hold eigrp 1 21
Device(config-if-vnet)# ip next-hop-self eigrp 1
! enable next-hop-self for e0/0.3
Device(config-if-vnet)# ip split-horizon eigrp 1
! enable split-horizon


Device(config-if-vnet)# do show running-config interface gigabitethernet 0/0/0

Building configuration...
Current configuration : 731 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
  vnet name vrf1
  ip split-horizon eigrp 1
  no ip authentication mode eigrp 1 md5
  ip authentication key-chain eigrp 1 y
  ip bandwidth-percent eigrp 1 99
  no ip dampening-change eigrp 1
  ip hello-interval eigrp 1 7
  ip hold-time eigrp 1 21
  !
end
```

Notice that g/0/0.3 is now using the override settings:

```
Device(config-if-vnet)# do show derived-config interface gigabitethernet 0/0.3

Building configuration...
Derived configuration : 479 bytes
!
interface GigabitEthernet0/0/0.3
```

```
 description Subinterface for VNET vrf1
 vrf forwarding vrf1
 encapsulation dot1Q 3
 ip address 192.0.2.1 255.255.255.0
 no ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 y
 ip bandwidth-percent eigrp 1 99
 no ip dampening-change eigrp 1
 ip hello-interval eigrp 1 7
 ip hold-time eigrp 1 21
 ip next-hop-self eigrp 1
 ip split-horizon eigrp 1
 end
```

Commands entered in virtual network interface mode are sticky. That is, when you enter a command in this mode, the command will override the default value configured in interface configuration mode.

The following example shows how to change the default hello interval value in vrf 1. The example also shows sample outputs of the current and derived configurations.

```
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# vnet trunk
Device(config-if)# ip hello eigrp 1 7
Device(config-if)# do show run interface gigabitethernet 0/0/2

Building configuration...
Current configuration : 134 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 7
 ipv6 enable
 vnet global
 !
 end

Device(config-if)# do show derived interface gigabitethernet 0/0/0.3

Building configuration...

Derived configuration : 177 bytes
!
interface Ethernet0/0.3
 description Subinterface for VNET vrf1
 encapsulation dot1Q 3
 vrf forwarding vrf1
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 7
end

Device(config-if)# vnet name vrf1
Device(config-if-vnet)# ip hello-interval eigrp 1 10
Device(config-if-vnet)# do show run interface gigabitethernet 0/0/0

Building configuration...
Current configuration : 183 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 7
```

```
 ipv6 enable
 vnet name vrf1
  ip hello-interval eigrp 1 10
 !
 vnet global
 !
end

Device(config-if-vnet)# do show derived interface gigabitethernet 0/0/0.3

Building configuration...

Derived configuration : 178 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET vrf1
 encapsulation dot1Q 3
 vrf forwarding vrf1
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 10
end
```

Because of this sticky factor, to remove a configuration entry in virtual network interface mode, use the default form of that command. Some commands can also be removed using the **no** form.

```
R1(config-if-vnet)# default ip authentication mode eigrp 1 md5
R1(config-if-vnet)# no ip bandwidth-percent eigrp 1
R1(config-if-vnet)# no ip hello eigrp 1

R1(config-if-vnet)# do show running-config interface gigabitethernet 0/0/0

Building configuration...
Current configuration : 138 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 no ip address
 vnet name vrf1
 !
end
```

Example: Monitoring and Maintaining the EIGRP Autonomous System Configuration

The **show ip eigrp** command displays prefix accounting information for EIGRP processes. The following is sample output from this command:

```
Device# show ip eigrp vrf VRF1 accounting

EIGRP-IPv4 Accounting for AS(100)/ID(10.0.2.1) VRF(VRF1)
Total Prefix Count: 4  States: A-Adjacency, P-Pending, D-Down
State Address/Source   Interface        Prefix   Restart  Restart/
                                        Count     Count   Reset(s)
 P    Redistributed     ----               0           3         211
 A    10.0.1.2          Gi0/0              2           0          84
 P    10.0.2.4          Se2/0              0           2         114
 D    10.0.1.3          Gi0/0              0           3           0
```

The **show ip eigrp events** command displays the EIGRP event log. The following is sample output from this command:

```
Device# show ip eigrp events

1    02:37:58.171 NSF stale rt scan, peer: 10.0.0.0
2    02:37:58.167 Metric set: 10.0.0.1/24 284700416
3    02:37:58.167 FC sat rdbmet/succmet: 284700416 0
4    02:37:58.167 FC sat nh/ndbmet: 10.0.0.2 284700416
5    02:37:58.167 Find FS: 10.0.0.0/24 284700416
6    02:37:58.167 Rcv update met/succmet: 284956416 284700416
7    02:37:58.167 Rcv update dest/nh: 10.0.0.0/24 10.0.0.1
8    02:37:58.167 Peer nsf restarted: 10.0.0.1 Tunnel0
9    02:36:38.383 Metric set: 10.0.0.0/24 284700416
10   02:36:38.383 RDB delete: 10.0.0.0/24 10.0.0.1
11   02:36:38.383 FC sat rdbmet/succmet: 284700416 0
12   02:36:38.383 FC sat nh/ndbmet: 0.0.0.0 284700416
```

The **show ip eigrp interfaces** command displays information about interfaces that are configured for EIGRP. The following is sample output from this command:

```
Device# show ip eigrp interfaces

EIGRP-IPv4 Interfaces for AS(60)
                    Xmit Queue    Mean   Pacing Time   Multicast   Pending
Interface   Peers   Un/Reliable   SRTT   Un/Reliable   Flow Timer  Routes
Gi0           0         0/0          0      11/434          0          0
Gi0           1         0/0        337       0/10           0          0
SE0:1.16      1         0/0         10       1/63         103          0
Tu0           1         0/0        330       0/16           0          0
```

The **show ip eigrp neighbors** command displays neighbors discovered by EIGRP. The following is sample output from this command:

```
Device# show ip eigrp neighbors

H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.1.2                 Gi0/0           13 00:00:03  1996   5000  0  5
2   10.1.1.9                 Gi0/0           14 00:02:24   206   5000  0  5
1   10.1.2.3                 Gi0/1           11 00:20:39  2202   5000  0  5
```

The **show ip eigrp topology** command displays entries in the EIGRP topology table. The following is sample output from this command:

```
Device# show ip eigrp topology

EIGRP-IPv4 Topology Table for AS(1)/ID(10.0.0.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status, s - sia status
P 10.0.0.0/8, 1 successors, FD is 409600
        via 10.0.0.1 (409600/128256), GigabirEthernet0/0
P 172.16.1.0/24, 1 successors, FD is 409600
        via 10.0.0.1 (409600/128256), GigabitEthernet0/0
P 10.0.0.0/8, 1 successors, FD is 281600
        via Summary (281600/0), Null0
P 10.0.1.0/24, 1 successors, FD is 281600
        via Connected, GigabitEthernet0/0
```

The **show ip eigrp traffic** command displays the number of EIGRP packets sent and received. The following is sample output from this command:

```
Device# show ip eigrp traffic

EIGRP-IPv4 Traffic Statistics for AS(60)
Hellos sent/received: 21429/2809
Updates sent/received: 22/17
Queries sent/received: 0/0
Replies sent/received: 0/0
Acks sent/received: 16/13
SIA-Queries sent/received: 0/0
SIA-Replies sent/received: 0/0
Hello Process ID: 204
PDM Process ID: 203
Socket Queue: 0/2000/2/0 (current/max/highest/drops)
Input Queue: 0/2000/2/0 (current/max/highest/drops)
```

Example: Monitoring and Maintaining the EIGRP Named Configuration

In this example, the **show eigrp address-family** command displays prefix accounting information for EIGRP processes:

```
Device# show eigrp address-family ipv4 22 accounting

EIGRP-IPv4 VR(saf) Accounting for AS(22)/ID(10.0.0.1)
Total Prefix Count: 3  States: A-Adjacency, P-Pending, D-Down
State Address/Source    Interface       Prefix   Restart  Restart/
                                        Count     Count   Reset(s)
 A    10.0.0.2          Gi0/0               2         0        0
 P    10.0.2.4          Se2/0               0         2        114
 D    10.0.1.3          Gi0/0               0         3        0
```

In this example, the **show eigrp address-family** command displays information about EIGRP address-family events:

```
Device# show eigrp address-family ipv4 3 events

Event information for AS 3:
1 15:37:47.015 Change queue emptied, entries: 1
2 15:37:47.015 Metric set: 10.0.0.0/24 307200
3 15:37:47.015 Update reason, delay: new if 4294967295
4 15:37:47.015 Update sent, RD: 10.0.0.0/24 4294967295
5 15:37:47.015 Update reason, delay: metric chg 4294967295
6 15:37:47.015 Update sent, RD: 10.0.0.0/24 4294967295
7 15:37:47.015 Route installed: 10.0.0.0/24 10.0.1.2
8 15:37:47.015 Route installing: 10.0.0.0/24 10.0.1.2
```

In this example, the **show eigrp address-family** command displays information about interfaces that are configured for EIGRP:

```
Device# show eigrp address-family ipv4 4453 interfaces

EIGRP-IPv4 VR(Virtual-name) Address-family Neighbors for AS(4453)
     Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface     Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Services
Se0           1          0/0        28      0/15          127           0
Se1           1          0/0        44      0/15          211           0
```

In this example, the **show eigrp address-family** command displays information about the neighbors that are discovered by EIGRP:

```
Device# show eigrp address-family ipv4 4453 neighbors
```

```
EIGRP-IPv4 VR(Virtual-name) Address-family Neighbors for AS(4453)
Address       Interface     Hold Uptime  SRTT  RTO    Q     Seq
                                 (sec)         (ms)   (ms)   Cnt   Num
172.16.81.28           GigabitEthernet1/1/1     13   0:00:41  0      11   4     20
172.16.80.28           GigabitEthernet0/0/1     14   0:02:01  0      10   12    24
172.16.80.31           GigabitEthernet0/1/1     12   0:02:02  0      4    5
```

In this example, the **show eigrp address-family** command displays information about EIGRP timers and expiration times:

```
Device# show eigrp address-family ipv4 4453 timers

EIGRP-IPv4 VR(Virtual-name) Address-family Timers for AS(4453)
Hello Process
Expiration Type
| 1.022 (parent)
| 1.022 Hello (Et0/0)
Update Process
Expiration Type
| 14.984 (parent)
| 14.984 (parent)
| 14.984 Peer holding
SIA Process
Expiration Type for Topo(base)
| 0.000 (parent)
```

In this example, the **show eigrp address-family** command displays entries in the EIGRP topology table:

```
Device# show eigrp address-family ipv4 4453 topology

EIGRP-IPv4 VR(Virtual-name) Topology Table for AS(4453)/ID(10.0.0.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status, s - sia Status
P  10.17.17.0/24, 1 successors, FD is 409600
          via 10.10.10.2 (409600/128256), GigabitEthernet3/0/1
P  172.16.19.0/24, 1 successors, FD is 409600
          via 10.10.10.2 (409600/128256), GigabitEthernet3/0/1
P  192.168.10.0/24, 1 successors, FD is 281600
          via Connected, GigabitEthernet3/0/1
P  10.10.10.0/24, 1 successors, FD is 281600
          via Redistributed (281600/0)
```

In this example, the **show eigrp address-family** command displays information about the number of EIGRP packets that are sent and received:

```
Device# show eigrp address-family ipv4 4453 traffic

EIGRP-IPv4 VR(virtual-name) Address-family Traffic Statistics for AS(4453)
  Hellos sent/received: 122/122
  Updates sent/received: 3/1
  Queries sent/received: 0/0
  Replies sent/received: 0/0
  Acks sent/received: 0/3
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 128
  PDM Process ID: 191
  Socket Queue: 0/2000/1/0 (current/max/highest/drops)
  Input Queue: 0/2000/1/0 (current/max/highest/drops
```

In this example, the **show eigrp plugins** command displays general information, including the versions of the EIGRP protocol features that are currently running on the device:

```
Device# show eigrp plugins

EIGRP feature plugins:::
    eigrp-release      :   5.00.00 : Portable EIGRP Release
                       :  19.00.00 : Source Component Release(rel5)
    igrp2              :   3.00.00 : Reliable Transport/Dual Database
    bfd                :   1.01.00 : BFD Platform Support
    mtr                :   1.00.01 : Multi-Topology Routing(MTR)
    eigrp-pfr          :   1.00.01 : Performance Routing Support
    ipv4-af            :   2.01.01 : Routing Protocol Support
    ipv4-sf            :   1.01.00 : Service Distribution Support
    external-client    :   1.02.00 : Service Distribution Client Support
    ipv6-af            :   2.01.01 : Routing Protocol Support
    ipv6-sf            :   1.01.00 : Service Distribution Support
    snmp-agent         :   1.01.01 : SNMP/SNMPv2 Agent Support
```

In this example, the **show eigrp protocols** command displays general information about EIGRP protocols that are currently running on a device:

```
Device# show eigrp protocols

EIGRP-IPv4 Protocol for AS(10)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 10.0.1.1
Topology : 0 (base)
Active Timer: 3 min
Distance: internal 90 external 170
Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1
EIGRP-IPv4 Protocol for AS(5) VRF(VRF1)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 10.1.2.1
Topology : 0 (base)
Active Timer: 3 min
Distance: internal 90 external 170
Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1
Total Prefix Count: 0
Total Redist Count: 0
```

Additional References for EIGRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
EIGRP commands	IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions

Related Topic	Document Title
EIGRP L2/L3 API and Tunable Metric for Mobile Adhoc Networks feature	"Mobile Ad Hoc Networks for Router-to-Radio Communications" module of *the IP Mobility Configuration Guide*
EIGRP Technology Support	Enhanced Interior Gateway Routing Protocol
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol
IPv6 Routing EIGRP Support	*IPv6 Routing: EIGRP Support*
Protocol-independent features that work with EIGRP	*IP Routing: Protocol-Independent Configuration Guide*
Service Advertisement Framework	*Service Advertisement Framework Configuration Guide*
Service Advertisement Framework commands	Service Advertisement Framework Command Reference

Standards and RFCs

Standard/RFC	Title
FIPS PUB 180-2	*SECURE HASH STANDARD (SHS)*
RFC 1321	*The MD5 Message-Digest Algorithm*
RFC 2104	*HMAC: Keyed-Hashing for Message Authentication*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 164: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 133

IPv6 Routing: EIGRP Support

Customers can configure Enhanced Interior Gateway Routing Protocol (EIGRP) to route IPv6 prefixes. EIGRP IPv4 runs over an IPv4 transport, communicates only with IPv4 peers, and advertises only IPv4 routes, and EIGRP for IPv6 follows the same model. EIGRP for IPv4 and EIGRP for IPv6 are configured and managed separately. However, the configuration of EIGRP for IPv4 and IPv6 is similar and provides operational familiarity and continuity.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for IPv6 Routing EIGRP Support

This section lists ways in which EIGRP for IPv6 differs from EIGRP IPv4 and lists EIGRP for IPv6 restrictions:

- EIGRP for IPv6 is directly configured on the interfaces over which it runs. This feature allows EIGRP for IPv6 to be configured without the use of a global IPv6 address. There is no network statement in EIGRP for IPv6.

In per-interface configuration at system startup, if EIGRP has been configured on an interface, then the EIGRP protocol may start running before any EIGRP router mode commands have been executed.

- An EIGRP for IPv6 protocol instance requires a router ID before it can start running.

- EIGRP for IPv6 has a shutdown feature. The routing process should be in "no shut" mode in order to start running.
- EIGRP for IPv6 provides route filtering using the **distribute-list prefix-list** command. Use of the **route-map** command is not supported for route filtering with a distribute list.

Information About IPv6 Routing EIGRP Support

Cisco EIGRP for IPv6 Implementation

EIGRP is an enhanced version of the IGRP developed by Cisco. EIGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm called the diffusing update algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize at the same time. Devices that are not affected by topology changes are not involved in recomputations. The convergence time with DUAL rivals that of any other existing routing protocol.

EIGRP provides the following features:

- Increased network width--With Routing Information Protocol (RIP), the largest possible width of your network is 15 hops. When EIGRP is enabled, the largest possible width is 224 hops. Because the EIGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this limitation by incrementing the transport control field only when an IPv4 or an IPv6 packet has traversed 15 devices and the next hop to the destination was learned by way of EIGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.
- Fast convergence--The DUAL algorithm allows routing information to converge as quickly as any other routing protocol.
- Partial updates--EIGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for EIGRP packets.
- Neighbor discovery mechanism--This is a simple hello mechanism used to learn about neighboring devices. It is protocol-independent.
- Arbitrary route summarization.
- Scaling--EIGRP scales to large networks.
- Route filtering--EIGRP for IPv6 provides route filtering using the **distribute-list prefix-list**command. Use of the **route-map** command is not supported for route filtering with a distribute list.

EIGRP has the following four basic components:

- Neighbor discovery--Neighbor discovery is the process that devices use to dynamically learn of other devices on their directly attached networks. Devices must also discover when their neighbors become unreachable or inoperative. EIGRP neighbor discovery is achieved with low overhead by periodically sending small hello packets. EIGRP neighbors can also discover a neighbor that has recovered after an

outage because the recovered neighbor will send out a hello packet. As long as hello packets are received, the Cisco software can determine that a neighbor is alive and functioning. Once this status is determined, the neighboring devices can exchange routing information.

- Reliable transport protocol--The reliable transport protocol is responsible for guaranteed, ordered delivery of EIGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some EIGRP packets must be sent reliably and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities, it is not necessary to send hello packets reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, which is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when unacknowledged packets are pending. This provision helps to ensure that convergence time remains low in the presence of varying speed links.
- DUAL finite state machine--The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses several metrics including distance and cost information to select efficient, loop-free paths. When multiple routes to a neighbor exist, DUAL determines which route has the lowest metric (named the feasible distance), and enters this route into the routing table. Other possible routes to this neighbor with larger metrics are received, and DUAL determines the reported distance to this network. The reported distance is defined as the total metric advertised by an upstream neighbor for a path to a destination. DUAL compares the reported distance with the feasible distance, and if the reported distance is less than the feasible distance, DUAL considers the route to be a feasible successor and enters the route into the topology table. The feasible successor route that is reported with the lowest metric becomes the successor route to the current route if the current route fails. To avoid routing loops, DUAL ensures that the reported distance is always less than the feasible distance for a neighbor device to reach the destination network; otherwise, the route to the neighbor may loop back through the local device.
- Protocol-dependent modules--When there are no feasible successors to a route that has failed, but there are neighbors advertising the route, a recomputation must occur. This is the process in which DUAL determines a new successor. The amount of time required to recompute the route affects the convergence time. Recomputation is processor-intensive; it is advantageous to avoid unneeded recomputation. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, DUAL will use them in order to avoid unnecessary recomputation.

The protocol-dependent modules are responsible for network layer protocol-specific tasks. For example, the EIGRP module is responsible for sending and receiving EIGRP packets that are encapsulated in IPv4 or IPv6. It is also responsible for parsing EIGRP packets and informing DUAL of the new information received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IPv4 or IPv6 routing table. Also, EIGRP is responsible for redistributing routes learned by other IPv4 or IPv6 routing protocols.

How to Configure IPv6 Routing EIGRP Support

Enabling EIGRP for IPv6 on an Interface

EIGRP for IPv6 is directly configured on the interfaces over which it runs, which allows EIGRP for IPv6 to be configured without the use of a global IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **no shut**
6. **ipv6 enable**
7. **ipv6 eigrp** *as-number*
8. **ipv6 router eigrp** *as-number*
9. **eigrp router-id** *router-id*
10. **exit**
11. **show ipv6 eigrp** [*as-number*] **interfaces** [*type number*] [*as-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface on which EIGRP is to be configured.
Step 5	**no shut** **Example:** `Device(config-if)# no shut`	Enables no shut mode so the routing process can start running.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

	Command or Action	Purpose
Step 7	**ipv6 eigrp** *as-number* **Example:** `Device(config-if)# ipv6 eigrp 1`	Enables EIGRP for IPv6 on a specified interface.
Step 8	**ipv6 router eigrp** *as-number* **Example:** `Device(config-if)# ipv6 router eigrp 1`	Enters router configuration mode and creates an EIGRP IPv6 routing process.
Step 9	**eigrp router-id** *router-id* **Example:** `Device(config-router)# eigrp router-id 10.1.1.1`	Enables the use of a fixed router ID. Use this command only if an IPv4 address is not defined on the router eligible for router ID.
Step 10	**exit** **Example:** `Device(config-router) exit`	Enter three times to return to privileged EXEC mode.
Step 11	**show ipv6 eigrp** [*as-number*] **interfaces** [*type number*] [*as-number*] **Example:** `Device# show ipv6 eigrp interfaces`	Displays information about interfaces configured for EIGRP for IPv6.

Configuring the Percentage of Link Bandwidth Used by EIGRP

By default, EIGRP packets consume a maximum of 50 percent of the link bandwidth, as configured with the **bandwidth interface** command. You might want to change that value if a different level of link utilization is required or if the configured bandwidth does not match the actual link bandwidth (it may have been configured to influence route metric calculations).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 bandwidth-percent eigrp** *as-number percent*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 bandwidth-percent eigrp** *as-number percent* **Example:** `Device(config-if)# ipv6 bandwidth-percent eigrp 1 75`	Configures the percentage of bandwidth that may be used by EIGRP for IPv6 on an interface

Configuring Summary Addresses

If any more specific routes are in the routing table, EIGRP for IPv6 will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 summary-address eigrp** *as-number ipv6-address* [*admin-distance*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 summary-address eigrp** *as-number ipv6-address* [*admin-distance*] **Example:** `Device(config-if)# ipv6 summary-address eigrp 1 2001:DB8:0:1::/64`	Configures a summary aggregate address for a specified interface.

Configuring EIGRP Route Authentication

EIGRP route authentication provides message digest algorithm 5 (MD5) authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

Each key has its own key identifier, which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and MD5 authentication key in use.

You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in order from lowest to highest, and uses the first valid key it encounters. Note that the router needs to know the time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 authentication mode eigrp** *as-number* **md5**
6. **ipv6 authentication key-chain eigrp** *as-number key-chain*
7. **exit**
8. **key chain** *name-of-chain*
9. **key** *key-id*

10. **key-string** *text*
11. **accept-lifetime** *start-time* **infinite** | *end-time*| **duration** *seconds*
12. **send-lifetime** *start-time* **infinite** | *end-time* | **duration** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 authentication mode eigrp** *as-number* **md5** **Example:** `Device(config-if)# ipv6 authentication mode eigrp 1 md5`	Specifies the type of authentication used in EIGRP for IPv6 packets.
Step 6	**ipv6 authentication key-chain eigrp** *as-number key-chain* **Example:** `Device(config-if)# ipv6 authentication key-chain eigrp 1 chain1`	Enables authentication of EIGRP for IPv6 packets.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	**key chain** *name-of-chain* **Example:** `Device(config)# key chain chain1`	Identifies a group of authentication keys. • Use the name specified in Step 5.

	Command or Action	Purpose
Step 9	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Identifies an authentication key on a key chain.
Step 10	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string chain 1`	Specifies the authentication string for a key.
Step 11	**accept-lifetime** *start-time* **infinite** | *end-time*| **duration** *seconds* **Example:** `Device(config-keychain-key)# accept-lifetime 14:30:00 Jan 10 2006 duration 7200`	Sets the time period during which the authentication key on a key chain is received as valid.
Step 12	**send-lifetime** *start-time* **infinite** | *end-time* | **duration** *seconds* **Example:** `Device(config-keychain-key)# send-lifetime 15:00:00 Jan 10 2006 duration 3600`	Sets the time period during which an authentication key on a key chain is valid to be sent.

Overriding the Next Hop in EIGRP

EIGRP will, by default, set the IPv6 next-hop value to be itself for routes that it is advertising, even when advertising those routes back out the same interface where it learned them. Perform this task to change this default and instruct EIGRP to use the received next-hop value when advertising these routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **no ipv6 next-hop-self eigrp** *as-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**no ipv6 next-hop-self eigrp** *as-number* **Example:** `Device(config-if)# no ipv6 next-hop-self eigrp 1`	Changes the default IPv6 next-hop value and instructs EIGRP to use the received next-hop value.

Adjusting the Interval Between Hello Packets in EIGRP for IPv6

Routing devices periodically send hello packets to each other to dynamically learn of other routers on their directly attached networks. This information is used to discover neighbors and to learn when neighbors become unreachable or inoperative.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 hello-interval eigrp** *as-number seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**ipv6 hello-interval eigrp** *as-number seconds* **Example:** `Device(config)# ipv6 hello-interval eigrp 1 10`	Configures the hello interval for the EIGRP for IPv6 routing process designated by an autonomous system number.

Adjusting the Hold Time in EIGRP for IPv6

On very congested and large networks, the default hold time might not be sufficient time for all routers to receive hello packets from their neighbors. In this case, you may want to increase the hold time.

Perform this task to configure the hold time on a specified interface for a particular EIGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds. For slow-speed nonbroadcast multi-access (NBMA) networks, the default hold time is 180 seconds. The hold time should be changed if the hello-interval value is changed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 hold-time eigrp** *as-number seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface on which EIGRP is configured.

	Command or Action	Purpose
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 hold-time eigrp** *as-number seconds* **Example:** `Device(config)# ipv6 hold-time eigrp 1 40`	Configures the hold time for a particular EIGRP for IPv6 routing process designated by the autonomous system number.

Disabling Split Horizon in EIGRP for IPv6

By default, split horizon is enabled on all interfaces. Split horizon controls the sending of EIGRP update and query packets. When split horizon is enabled on an interface, update and query packets are not sent for destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

Split horizon blocks route information from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks (such as multipoint GRE), situations can arise for which this behavior is not ideal. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **no ipv6 split-horizon eigrp** *as-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies the interface on which EIGRP is configured.

	Command or Action	Purpose
	`Device(config)# interface GigabitEthernet 0/0/0`	
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**no ipv6 split-horizon eigrp** *as-number* **Example:** `Device(config-if)# no ipv6 split-horizon eigrp 101`	Disables EIGRP for IPv6 split horizon on the specified interface.

Configuring EIGRP Stub Routing for Greater Network Stability

The EIGRP stub routing feature can help to provide greater network stability. In the event of network instability, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit devices. Instead, distribution devices to which the stub device is connected answer the query on behalf of the stub device. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP stub routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote devices to prevent those remote devices from appearing as transit paths to the hub devices.

Caution EIGRP stub routing should be used only on stub devices. A stub device is defined as a device connected to the network core or distribution layer through which core transit traffic should not flow. A stub device should not have any EIGRP neighbors other than distribution devices.

Configuring a Device for EIGRP Stub Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **eigrp stub receive-only** | **leak-map** | **connected** | **static** | **summary** | **redistributed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**eigrp stub receive-only** | **leak-map** | **connected** | **static** | **summary** | **redistributed** **Example:** `Device(config-router)# eigrp stub`	Configures a device as a stub using EIGRP.

Verifying EIGRP Stub Routing

SUMMARY STEPS

1. **enable**
2. **show ipv6 eigrp neighbors detail** *interface-type* | *as-number* | **static**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 eigrp neighbors detail** *interface-type* | *as-number* | **static** **Example:** `Device# show ipv6 eigrp neighbors detail`	Displays the neighbors discovered by EIGRP for IPv6. This command is performed on the distribution layer device to view the status of the remote device.

Customizing an EIGRP for IPv6 Routing Process

Logging EIGRP Neighbor Adjacency Changes

You can enable the logging of neighbor adjacency changes to monitor the stability of the routing system and to help you detect problems. By default, adjacency changes are logged.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **eigrp log-neighbor-changes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**eigrp log-neighbor-changes** **Example:** `Device(config-router)# eigrp log-neighbor-changes`	Enables the logging of changes in EIGRP for IPv6 neighbor adjacencies.

Configuring Intervals Between Neighbor Warnings

When neighbor warning messages occur, they are logged by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **eigrp log-neighbor-warnings** [*seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**eigrp log-neighbor-warnings** [*seconds*] **Example:** `Device(config-router)# eigrp log-neighbor-warnings 300`	Configures the logging intervals of EIGRP neighbor warning messages.

Adjusting EIGRP for IPv6 Metric Weights

EIGRP for IPv6 uses the minimum bandwidth on the path to a destination network and the total delay to compute routing metrics. You can use the **metric weights** command to adjust the default behavior of EIGRP for IPv6 routing and metric computations. EIGRP for IPv6 metric defaults have been carefully selected to provide optimal performance in most networks.

Note Adjusting EIGRP metric weights can dramatically affect network performance. Because of the complexity of this task, we recommend that you do not change the default values without guidance from an experienced network designer.

By default, the EIGRP composite metric is a 32-bit quantity that is a sum of the segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (e.g., GigabitEthernet, FastEthernet, Ethernet), the route with the lowest metric reflects the most desirable path to a destination.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **metric weights** *tos k1 k2 k3 k4 k5*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**metric weights** *tos k1 k2 k3 k4 k5* **Example:** `Device(config-router)# metric weights 0 2 0 2 0 0`	Tunes EIGRP metric calculations.

Deleting Entries from EIGRP for IPv6 Routing Tables

SUMMARY STEPS

1. **enable**
2. **clear ipv6 eigrp** [*as-number*] [**neighbor** [*ipv6-address* | *interface-type interface-number*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 eigrp** [*as-number*] [**neighbor** [*ipv6-address* | *interface-type interface-number*]] **Example:** `Device# clear ipv6 eigrp neighbor 3FEE:12E1:2AC1:EA32`	Deletes entries from EIGRP for IPv6 routing tables. The routes that are cleared are the routes that were learned by the specified device.

Configuration Examples for IPv6 Routing EIGRP Support

Example: Configuring EIGRP to Establish Adjacencies on an Interface

EIGRP for IPv6 is configured directly on the interfaces over which it runs. This example shows the minimal configuration required for EIGRP for IPv6 to send hello packets in order to establish adjacencies on GigabitEthernet 0/0/0:

```
ipv6 unicast-routing
interface gigabitethernet0/0/0
no shut
  ipv6 enable
  ipv6 eigrp 1
!
ipv6 router eigrp 1
  eigrp router-id 10.1.1.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
CEF commands	*Cisco IOS IP Switching Command Reference*
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
NSF with SSO deployment	Cisco Nonstop Forwarding with Stateful Switchover Deployment Guide

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 4724	*Graceful Restart Mechanism for BGP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 165: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 134

EIGRP MPLS VPN PE-CE Site of Origin

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces the capability to filter Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) traffic on a per-site basis for Enhanced Interior Gateway Routing Protocol (EIGRP) networks. Site of Origin (SoO) filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent transient routing loops from occurring in complex and mixed network topologies. This feature is designed to support the MPLS VPN Support for EIGRP Between Provider Edge (PE) and Customer Edge (CE) feature. Support for backdoor links is provided by this feature when installed on PE routers that support EIGRP MPLS VPNs.

Prerequisites for EIGRP MPLS VPN PE-CE Site of Origin

This document assumes that Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone). The following tasks will also need to be completed before you can configure this feature:

- This feature was introduced to support the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature and should be configured after the EIGRP MPLS VPN is created.
- All PE routers that are configured to support the EIGRP MPLS VPN must run Cisco IOS XE Release 2.1 or a later release, which provides support for the SoO extended community.

Restrictions for EIGRP MPLS VPN PE-CE Site of Origin

- If a VPN site is partitioned and the SoO extended community attribute is configured on a backdoor router interface, the backdoor link cannot be used as an alternate path to reach prefixes originated in other partitions of the same site.

- A unique SoO value must be configured for each individual VPN site. The same value must be configured on all provider edge and customer edge interfaces (if SoO is configured on the CE routers) that support the same VPN site.

Information About EIGRP MPLS VPN PE-CE Site of Origin

EIGRP MPLS VPN PE-CE Site of Origin Support Overview

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces SoO support for EIGRP-to-BGP and BGP-to-EIGRP redistribution. The SoO extended community is a BGP extended community attribute that is used to identify routes that have originated from a site so that the readvertisement of that prefix back to the source site can be prevented. The SoO extended community uniquely identifies the site from which a PE router has learned a route. SoO support provides the capability to filter MPLS VPN traffic on a per-EIGRP-site basis. SoO filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent routing loops from occurring in complex and mixed network topologies, such as EIGRP VPN sites that contain both VPN and backdoor links.

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

Site of Origin Support for Backdoor Links

The EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature introduces support for backdoor links. A backdoor link or a route is a connection that is configured outside of the VPN between a remote and main site; for example, a WAN leased line that connects a remote site to the corporate network. Backdoor links are typically used as back up routes between EIGRP sites if the VPN link is down or not available. A metric is set on the backdoor link so that the route though the backdoor router is not selected unless there is a VPN link failure.

The SoO extended community is defined on the interface of the backdoor router. It identifies the local site ID, which should match the value that is used on the PE routers that support the same site. When the backdoor router receives an EIGRP update (or reply) from a neighbor across the backdoor link, the router checks the update for an SoO value. If the SoO value in the EIGRP update matches the SoO value on the local backdoor interface, the route is rejected and not added to the EIGRP topology table. This scenario typically occurs when the route with the local SoO valued in the received EIGRP update was learned by the other VPN site and then advertised through the backdoor link by the backdoor router in the other VPN site. SoO filtering on the backdoor link prevents transient routing loops from occurring by filtering out EIGRP updates that contain routes that carry the local site ID.

Note If a VPN site is partitioned and the SoO extended community attribute is configured on a backdoor router interface, the backdoor link cannot be used as an alternate path to reach prefixes originated in other partitions of the same site.

If this feature is enabled on the PE routers and the backdoor routers in the customer sites, and SoO values are defined on both the PE and backdoor routers, both the PE and backdoor routers will support convergence

between the VPN sites. The other routers in the customer sites need only propagate the SoO values carried by the routes, as the routes are forwarded to neighbors. These routers do not otherwise affect or support convergence beyond normal Diffusing Update Algorithm (DUAL) computations.

Router Interoperation with the Site of Origin Extended Community

The configuration of an SoO extended community allows routers that support EIGRP MPLS VPN PE-CE Site of Origin feature to identify the site from which each route originated. When this feature is enabled, the EIGRP routing process on the PE or CE router checks each received route for the SoO extended community and filters based on the following conditions:

- A received route from BGP or a CE router contains an SoO value that matches the SoO value on the receiving interface.

If a route is received with an associated SoO value that matches the SoO value that is configured on the receiving interface, the route is filtered because it was learned from another PE router or from a backdoor link. This behavior is designed to prevent routing loops.

- A received route from a CE router is configured with an SoO value that does not match.

If a route is received with an associated SoO value that does not match the SoO value that is configured on the receiving interface, the route is added to the EIGRP topology table so that it can be redistributed into BGP.

If the route is already installed to the EIGRP topology table but is associated with a different SoO value, the SoO value from the topology table will be used when the route is redistributed into BGP.

- A received route from a CE router does not contain an SoO value.

If a route is received without a SoO value, the route is accepted into the EIGRP topology table, and the SoO value from the interface that is used to reach the next hop CE router is appended to the route before it is redistributed into BGP.

When BGP and EIGRP peers that support the SoO extended community receive these routes, they will also receive the associated SoO values and pass them to other BGP and EIGRP peers that support the SoO extended community. This filtering is designed to prevent transient routes from being relearned from the originating site, which prevents transient routing loops from occurring.

Redistribution of BGP VPN Routes That Carry the Site of Origin into EIGRP

When an EIGRP routing process on a PE router redistributes BGP VPN routes into an EIGRP topology table, EIGRP extracts the SoO value (if one is present) from the appended BGP extended community attributes and appends the SoO value to the route before adding it to the EIGRP topology table. EIGRP tests the SoO value for each route before sending updates to CE routers. Routes that are associated with SoO values that match the SoO value configured on the interface are filtered out before they are passed to the CE routers. When an EIGRP routing process receives routes that are associated with different SoO values, the SoO value is passed to the CE router and carried through the CE site.

BGP Cost Community Support for EIGRP MPLS VPN PE-CE Network Topologies

The BGP cost community is a nontransitive extended community attribute that is passed to internal BGP (iBGP) and confederation peers but not external BGP (eBGP) peers. The cost community feature allows you to customize the local route preference and influence the BGP best path selection process.

Before BGP cost community support for EIGRP MPLS VPN PE-CE network topologies was introduced, BGP preferred locally sourced routes over routes learned from BGP peers. Backdoor links in an EIGRP MPLS VPN topology were preferred by BGP when the backdoor link was learned first. (A backdoor link or a route is a connection that is configured outside of the VPN between a remote and main site; for example, a WAN leased line that connects a remote site to the corporate network).

The "prebest path" point of insertion (POI) was introduced in the BGP Cost Community feature to support mixed EIGRP VPN network topologies that contain VPN and backdoor links. This POI is applied automatically to EIGRP routes that are redistributed into BGP. The "prebest path" POI carries the EIGRP route type and metric. This POI influences the best path calculation process by influencing BGP to consider this POI before any other comparison step. No configuration is required. This feature is enabled automatically for EIGRP VPN sites when Cisco IOS XE Release 2.1 or later is installed on the PE routers or the CE and backdoor router at the customer sites.

For more information about the BGP Cost Community feature, see to the BGP Cost Community module in the *Cisco IOS XE IP Routing: BGP Configuration Guide, Release 2*.

Benefits of the EIGRP MPLS VPN PE-CE Site of Origin Support Feature

The configuration of the EIGRP MPLS VPN PE-CE Site of Origin Support feature introduces per-site VPN filtering, which improves support for complex topologies, such as MPLS VPNs with backdoor links, CE routers that are dual-homed to different PE routers, and PE routers that support CE routers from different sites within the same virtual routing and forwarding (VRF) instance.

How to Configure EIGRP MPLS VPN PE-CE Site of Origin Support

Configuring the Site of Origin Extended Community

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

Before you begin

- Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone).
- Configure an EIGRP MPLS VPN before configuring this feature.
- All PE routers that are configured to support the EIGRP MPLS VPN must support the SoO extended community.
- A unique SoO value must be configured for each VPN site. The same value must be used on the interface of the PE router that connects to the CE router for each VPN site.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
4. **set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
5. **exit**
6. **interface** *type number*
7. **ip vrf forwarding** *vrf-name*
8. **ip vrf sitemap** *route-map-name*
9. **ip address** *ip-address subnet-mask*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-name* {**permit**	**deny**} [*sequence-number*] **Example:** `Router(config)# route-map Site-of-Origin permit 10`	Enters route-map configuration mode and creates a route map. • The route map is created in this step so that SoO extended community can be applied.

Step 4	**set extcommunity** {**rt** *extended-community-value* [**additive**]	**soo** *extended-community-value*}
		Example:
		`Router(config-route-map)# set extcommunity soo 100:1`
			Sets BGP extended community attributes.
			• The **rt** keyword specifies the route target extended community attribute.
			• The **soo** keyword specifies the site of origin extended community attribute.
			• The *extended-community-value*argument specifies the value to be set. The value can be one of the following formats:
			  • autonomous-system-number: network-number
			  • ip-address: network-number
			The colon is used to separate the autonomous system number and network number or IP address and network number.
			• The **additive** keyword adds a route target to the existing route target list without replacing any existing route targets.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 0/0`	Enters interface configuration mode to configure the specified interface.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding VRF1`	Associates the VRF with an interface or subinterface. • The VRF name configured in this step should match the VRF name created for the EIGRP MPLS VPN with the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature.
Step 8	**ip vrf sitemap** *route-map-name* **Example:** `Router(config-if)# ip vrf sitemap Site-of-Origin`	Associates the VRF with an interface or subinterface. • The route map name configured in this step should match the route map name created to apply the SoO extended community in Step 3.
Step 9	**ip address** *ip-address subnet-mask* **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.255`	Configures the IP address for the interface. • The IP address needs to be reconfigured after enabling VRF forwarding.
Step 10	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

What to Do Next

- For mixed EIGRP MPLS VPN network topologies that contain backdoor routes, the next task is to configure the “prebest path” cost community for backdoor routes.

Verifying the Configuration of the SoO Extended Community

Use the following steps to verify the configuration of the SoO extended community attribute.

SUMMARY STEPS

1. **enable**

2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher*| **vrf** *vrf-name*} [*ip-prefix*/*length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher*| **vrf** *vrf-name*} [*ip-prefix*/*length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**] **Example:** `Router# show ip bgp vpnv4 all 10.0.0.1`	Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4** command with the **all** keyword to verify that the specified route has been configured with the SoO extended community attribute.

Configuration Examples for EIGRP MPLS VPN PE-CE SoO

Example Configuring the Site of Origin Extended Community

The following example, beginning in global configuration mode, configures SoO extended community on an interface:

```
Router(config)# route-map Site-of-Origin permit 10

Router(config-route-map)# set extcommunity soo 100:1
Router(config-route-map)# exit

Router(config)# interface FastEthernet 0/0

Router(config-if)# ip vrf forwarding RED
Router(config-if)# ip vrf sitemap Site-of-Origin
Router(config-if)# ip address 10.0.0.1 255.255.255.255
Router(config-if)# end
```

Example Verifying the Site of Origin Extended Community

The following example shows VPN address information from the BGP table and verifies the configuration of the SoO extended community:

```
Router# show ip bgp vpnv4 all 10.0.0.1
BGP routing table entry for 100:1:10.0.0.1/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  100 300
    192.168.0.2 from 192.168.0.2 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: SOO:100:1
```

The following example shows how to display EIGRP metrics for specified internal services and external services:

```
Router# show eigrp address-family ipv4 4453 topology 10.10.10.0/24
EIGRP-IPv4 VR(virtual-name) Topology Entry for AS(4453)/ID(10.0.0.1) for 10.10.10.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Descriptor Blocks:
  0.0.0.0 (Null0), from Connected, Send flag is 0x0
      Composite metric is (128256/0), service is Internal
      Vector metric:
        Minimum bandwidth is 10000000 Kbit
        Total delay is 5000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1514
        Hop count is 0
        Originating router is 10.0.0.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP Cost Community feature and the “pre-bestpath” point of insertion	BGP Cost Community module of the *Cisco IOS IP Routing: BGP Configuration Guide*
CEF commands	*Cisco IOS IP Switching Command Reference*
CEF configuration tasks	Cisco Express Forwarding Overview module of the *Cisco IOS IP Switching Configuration Guide*
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
EIGRP configuration tasks	Configuring EIGRP
MPLS VPNs	MPLS Layer 3 VPNs module of the *Cisco IOS Multiprotocol Label Switching Configuration Guide*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 166: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

Glossary

AFI --Address Family Identifier. Carries the identity of the network layer protocol that is associated with the network address.

Backdoor link --A link connecting two backdoor routers.

Backdoor router --A router that connects two or more sites, that are also connected to each other through an MPLS VPN EIGRP PE to CE links.

BGP --Border Gateway Protocol. An interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined by RFC 1163, A Border Gateway Protocol (BGP). BGP supports CIDR and uses route aggregation mechanisms to reduce the size of routing tables.

Cost Community --An extended community attribute that can be inserted anywhere into the best path calculation.

customer edge (CE) router --A router that belongs to a customer network, that connects to a provider edge (PE) router to utilize MPLS VPN network services.

MBGP --multiprotocol BGP. An enhanced version of BGP that carries routing information for multiple network-layer protocols and IP multicast routes. It is defined in RFC 2858, Multiprotocol Extensions for BGP-4.

provider edge (PE) router --The PE router is the entry point into the service provider network. The PE router is typically deployed on the edge of the network and is administered by the service provider. The PE router is the redistribution point between EIGRP and BGP in PE to CE networking.

site --A collection of routers that have well-defined exit points to other “sites.”

site of origin (SoO) --A special purpose tag or attribute that identifies the site that injects a route into the network. This attribute is used for intersite filtering in MPLS VPN PE-to-CE topologies.

VPN --Virtual Private Network. Allows IP traffic to travel securely over public TCP/IP networks and the Internet by encapsulating and encrypting all IP packets. VPN uses a tunnel to encrypt all information at the IP level.

CHAPTER 135

EIGRP Nonstop Forwarding Awareness

Nonstop Forwarding (NSF) awareness allows an NSF-aware router to assist NSF-capable and NSF-aware neighbors to continue forwarding packets during a switchover operation or during a well-known failure condition. The EIGRP Nonstop Forwarding Awareness feature allows an NSF-aware router that is running Enhanced Interior Gateway Routing Protocol (EIGRP) to forward packets along routes that are already known for a router that is performing a switchover operation or is in a well-known failure mode. This capability allows the EIGRP peers of the failing router to retain the routing information that is advertised by the failing router and continue to use this information until the failed router has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation.

Prerequisites for EIGRP Nonstop Forwarding Awareness

This module assumes that your network is configured to run EIGRP. The following tasks must also be completed before you can configure this feature:

- An NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in an NSF restart operation.
- A version of Cisco software that supports NSF awareness or NSF capabilities must be installed.
- Enable NSF on both DT and peer nodes during stacking.

Restrictions for EIGRP Nonstop Forwarding Awareness

- All neighboring devices that are participating in EIGRP NSF must be NSF-capable or NSF-aware.

- EIGRP NSF awareness does not support two neighbors that are performing an NSF restart operation at the same time. However, both neighbors will still re-establish peering sessions after the NSF restart operation is complete.

Information About EIGRP Nonstop Forwarding Awareness

Cisco NSF Routing and Forwarding Operation

Cisco NSF is supported by the BGP, EIGRP, OSPF, and IS-IS protocols for routing and by Cisco Express Forwarding (CEF) for forwarding. Of the routing protocols, BGP, EIGRP, OSPF, and IS-IS have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices. The IS-IS protocol can be configured to use state information that has been synchronized between the active and the standby RP to recover route information following a switchover instead of information received from peer devices.

In this document, a networking device is said to be NSF-aware if it is running NSF-compatible software. A device is said to be NSF-capable if it has been configured to support NSF; therefore, it would rebuild routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. Once the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF, in turn, updates the line cards with the new FIB information.

Note NSF supports IPv4 in classic mode and named mode. NSF supports IPv6 in named mode. For more information about EIGRP IPv6 NSF, see the "EIGRP IPv6 NSF/GR" module in the *IP Routing: EIGRP Configuration Guide*.

Cisco Express Forwarding

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by CEF. CEF maintains the FIB, and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active RP synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby RP. Upon switchover of the active RP, the standby RP initially has FIB and adjacency databases that are mirror images of those that were current on the active RP. For platforms with intelligent line cards, the line cards will maintain the current forwarding information over a switchover; for platforms with forwarding engines, CEF will keep the forwarding engine on the standby RP current with changes that are sent to it by CEF on the active RP. In this way, the line cards or forwarding engines will be able to continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates in turn cause prefix-by-prefix updates for CEF, which it uses to update the FIB and adjacency databases. Existing and new entries will receive the new version ("epoch") number, indicating that they have been refreshed. The forwarding information is updated on the line cards or forwarding engine during convergence. The RP signals when the

RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

The routing protocols run only on the active RP, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby RP. Following a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables.

Note For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.

EIGRP Nonstop Forwarding Awareness

NSF awareness allows a router that is running EIGRP to assist NSF-capable neighbors to continue forwarding packets during a switchover operation or well-known failure condition. The EIGRP Nonstop Forwarding Awareness feature provides EIGRP with the capability to detect a neighbor that is undergoing an NSF restart event (route processor [RP] switchover operation) or well-known failure condition, to maintain the peering session with this neighbor, to retain known routes, and to continue to forward packets for these routes. The deployment of EIGRP NSF awareness can minimize the effects of the following:

- Well-known failure conditions (for example, a stuck-in-active event).
- Unexpected events (for example, an RP switchover operation).
- Scheduled events (for example, a hitless software upgrade).

EIGRP NSF awareness is enabled by default, and its operation is transparent to the network operator and EIGRP peers that do not support NSF capabilities.

Note An NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in an NSF restart operation.

EIGRP NSF-Capable and NSF-Aware Interoperation

EIGRP NSF capabilities are exchanged by EIGRP peers in hello packets. The NSF-capable router notifies its neighbors that an NSF restart operation has started by setting the restart (RS) bit in a hello packet. When an NSF-aware router receives notification from an NSF-capable neighbor that an NSF-restart operation is in progress, the NSF-capable and NSF-aware routers immediately exchange their topology tables. The NSF-aware router sends an end-of-table (EOT) update packet when the transmission of its topology table is complete. The NSF-aware router then performs the following actions to assist the NSF-capable router:

- The router expires the EIGRP hello hold timer to reduce the time interval set for hello packet generation and transmission. This allows the NSF-aware router to reply to the NSF-capable router more quickly and reduces the amount of time required for the NSF-capable router to rediscover neighbors and rebuild the topology table.
- The router starts the graceful-restart purge-time timer. This timer is used to set the period of time that the NSF-aware router will hold known routes for the NSF-capable neighbor. This timer is configured with the **timers graceful-restart purge-time** command. The default time period is 240 seconds.

- The router notes in the peer list that the NSF-capable neighbor is restarting, maintains adjacency, and holds known routes for the NSF-capable neighbor until the neighbor signals that it is ready for the NSF-aware router to send its topology table or the graceful-restart purge-time timer expires. If the graceful-restart purge-time timer expires on the NSF-aware router, the NSF-aware router will discard held routes and treat the NSF-capable router as a new router joining the network and reestablishing adjacency accordingly.

When the switchover operation is complete, the NSF-capable router notifies its neighbors that it has reconverged and has received all of their topology tables by sending an EOT update packet to the assisting routers. The NSF-capable then returns to normal operation. The NSF-aware router will look for alternate paths (go active) for any routes that are not refreshed by the NSF-capable (restarting router). The NSF-aware router will then return to normal operation. If all paths are refreshed by the NSF-capable router, the NSF-aware router will immediately return to normal operation.

Non-NSF Aware EIGRP Neighbors

NSF-aware routers are completely compatible with non-NSF aware or capable neighbors in an EIGRP network. A non-NSF aware neighbor will ignore NSF capabilities and reset the adjacency when they are received.

The NSF-capable router will drop any queries that are received while converging to minimize the number of transient routes that are sent to neighbors. But the NSF-capable router will still acknowledge these queries to prevent these neighbors from resetting adjacency.

Note NSF-aware router will continue to send queries to the NSF-capable router which is still in the process of converging after switchover, effectively extending the time before a stuck-in-active (SIA) condition can occur.

EIGRP NSF Timers

NSF/GR supports three types of timers: namely, signal timer, converge timer, and graceful-restart purge-time timer.

The signal timer can be configured to adjust the maximum time of the initial restart period where the restarting router sends hello packets with the restart(RS)-bit set. When the timer expires, if the restarting router has not learnt about any neighbor, or has not learnt about any NSF-aware neighbor, or has not received all the updates from the neighbors, the routing information base is notified for convergence. The default value for the signal timer is 20 seconds. The **timers nsf signal** command is used to configure the signal timer.

The converge timer can be configured to adjust the maximum time the restarting router waits for the end-of-table (EOT) indications from all the neighbors. The default value for the converge timer is 120 seconds. The **timers nsf converge** command is used to configure the converge timer.

The graceful-restart purge-time timer can be configured to adjust the maximum waiting time to receive the convergent signal from the restarting router. The graceful-restart purge-timer is used when the NSF-aware peer does not receive the EOT indication from the restarting neighbor. When the graceful-restart purge-timer expires, the EIGRP peer scans the topology table for the stale routes from the restarting neighbor and changes the stale routes to active, thereby allowing EIGRP peers to find alternate routes instead of waiting during a long switchover operation. The default value for the graceful-restart purge-time timer is 240 seconds. The **timers graceful-restart purge-time** command is used to configure the graceful-restart purge-timer. The **timers graceful-restart purge-time** command is accepted under router configuration mode for IPv4 EIGRP classic mode and under address-family configuration mode for EIGRP named mode.

How to Configure EIGRP Nonstop Forwarding Awareness

Enabling EIGRP Nonstop Forwarding Awareness

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv4 autonomous-system** *number*
5. **nsf**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Configures an EIGRP routing process in classic mode and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 1`	Enters address-family configuration mode to configure an EIGRP routing instance.
Step 5	**nsf** **Example:** `Device(config-router-af)# nsf`	Enables NSF for the specific address family on the router.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address-family configuration mode and returns to privileged EXEC mode.

Modifying EIGRP Nonstop Forwarding Awareness Timers

Perform this task to modify EIGRP NSF timers. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *name*
4. **address-family ipv4 autonomous-system** *number*
5. **timers nsf signal** *seconds*
6. **timers nsf converge** *seconds*
7. **timers graceful-restart purge-time** *seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *name* **Example:** `Device(config)# router eigrp e1`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 1`	Enters address-family configuration mode to configure an EIGRP routing instance.
Step 5	**timers nsf signal** *seconds* **Example:** `Device(config-router-af)# timers nsf signal 15`	Sets the initial restart period wherein the restarting router sends hello packets with the RS-bit set. The default is 20 seconds.

	Command or Action	Purpose
Step 6	**timers nsf converge** *seconds* **Example:** `Device(config-router-af)# timers nsf converge 60`	Sets the maximum time that the restarting router has to wait for the EOT indications from all neighbors. The default is 120 seconds.
Step 7	**timers graceful-restart purge-time** *seconds* **Example:** `Device(config-router-af)# timers graceful-restart purge-time 150`	Sets the graceful-restart purge time to determine the period for which an NSF-aware router that is running EIGRP will hold routes for an inactive peer. The default is 240 seconds.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address-family configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear ip eigrp neighbors** command will need to be entered before normal redistribution will occur.

Monitoring EIGRP NSF Debug Events and Notifications

Use the following steps to monitor EIGRP NSF debug events and notifications on an NSF-aware router.

The **debug eigrp nsf** and **debug ip eigrp notifications** commands do not need to be issued together or even in the same session because there are differences in the information that is provided. These commands are provided together for example purposes.

The output of **debug** commands can be very verbose. These commands should not be deployed in a production network unless you are troubleshooting a problem.

SUMMARY STEPS

1. **enable**
2. **debug eigrp nsf**
3. **debug ip eigrp notifications**
4. **debug eigrp address-family ipv4 notifications**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**debug eigrp nsf** **Example:** `Device# debug eigrp nsf`	Displays NSF notifications and information about NSF events in an EIGRP network on the console of the router.
Step 3	**debug ip eigrp notifications** **Example:** `Device# debug ip eigrp notifications`	Displays EIGRP events and notifications in the console of the router. The output from this command also includes NSF notifications and information about NSF events.
Step 4	**debug eigrp address-family ipv4 notifications** **Example:** `Device# debug eigrp address-family ipv4 notifications`	Displays debugging information about EIGRP address-family IPv4 event notifications.

Verifying the Local Configuration of EIGRP NSF Awareness

Use the following steps to verify the local configuration of NSF-awareness on a router that is running EIGRP:

SUMMARY STEPS

1. **enable**
2. **show ip protocols**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip protocols** **Example:** `Device# show ip protocols`	Displays the parameters and current state of the active routing protocol process. The output of this command can be used to verify EIGRP NSF-awareness.

Configuration Examples for EIGRP Nonstop Forwarding Awareness

Example: EIGRP Graceful-Restart Purge-Time Timer Configuration

The following example shows how to set the graceful-restart purge-time timer to 2 minutes:

```
Device(config-router)# timers graceful-restart purge-time 120
```

Example: Monitoring EIGRP NSF Debug Events and Notifications Configuration

The following example output shows that an NSF-aware router has received a restart notification. The NSF-aware router waits for EOT to be sent from the restarting (NSF-capable) neighbor.

```
Device# debug ip eigrp notifications

*Oct  4 11:39:18.092:EIGRP:NSF:AS2. Rec RS update from 10.100.10.1,
00:00:00. Wait for EOT.
*Oct  4 11:39:18.092:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 2:Neighbor
10.100.10.1 (POS3/0) is up:peer NSF restarted
*Sep 23 18:49:07.578: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 1.1.2.1
(GigabitEthernet1/0/0) is resync: peer graceful-restart
```

Example: Verifying Local Configuration of EIGRP NSF Awareness

The following is example output from the **show ip protocols** command. The output from this command can be used to verify the local configuration of the EIGRP NSF awareness. The output below shows that the router is NSF-aware and that the graceful-restart purge-time timer is set to 240 seconds, which is the default value.

```
Device# show ip protocols

*** IP Routing is NSF aware ***
Routing Protocol is "eigrp 101"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 101
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.4.9.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

Additional References for EIGRP Nonstop Forwarding Awareness

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
CEF commands	*Cisco IOS IP Switching Command Reference*
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
Nonstop forwarding (NSF)	• Cisco Nonstop Forwarding with Stateful Switchover Deployment Guide • "Cisco Nonstop Forwarding" module in *High Availability Configuration Guide* • "EIGRP IPv6 NSF/GR" module in *IP Routing: EIGRP Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 167: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 136

EIGRP Nonstop Forwarding

EIGRP nonstop forwarding (NSF) capabilities are exchanged by EIGRP peers in hello packets. NSF works with the SSO feature in Cisco software to minimize the amount of time that a network is unavailable to its users following a switchover. The main objective of NSF is to continue forwarding IP packets following a Route Processor (RP) switchover.

Note Throughout this document, the term Route Processor (RP) is used to describe the route processing engine on all networking devices, regardless of the platform designation, unless otherwise noted.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for EIGRP Nonstop Forwarding

- The networking device that is to be configured for NSF must first be configured for SSO. For more information, see the “Configuring Stateful Switchover” chapter in the *High Availability Configuration Guide*.
- All neighboring devices must be NSF-capable or NSF-aware.

- An NSF-aware device must be completely converged with the network before it can assist an NSF-capable device in an NSF restart operation.
- On platforms that support the Route Switch Processor (RSP), and where the Cisco Express Forwarding (CEF) switching mode is configurable, configure distributed CEF (dCEF) switching mode using the **ip cef distributed** command.

Note Distributed platforms that run a supporting version of Cisco software can support full NSF capabilities. These devices can perform a restart operation and can support other NSF capable peers.

Restrictions for EIGRP Nonstop Forwarding

- An NSF-aware device cannot support two NSF-capable peers that are performing an NSF restart operation at the same time. However, both neighbors will reestablish peering sessions after the NSF restart operation is complete.
- Single processor platforms that run a supporting version of Cisco software support only NSF awareness. These devices maintain adjacency and hold known routes for the NSF-capable neighbor until it signals that it is ready for the NSF-aware device to send its topology table or until the route-hold timer expires.

Information About EIGRP Nonstop Forwarding

Nonstop Forwarding

Note In the following content, the term Route Processor (RP) is used to describe the route processing engine on all networking devices, regardless of the platform designation, unless otherwise noted.

NSF works with the SSO feature in Cisco software to minimize the amount of time a network is unavailable to its users following a switchover. The main objective of NSF is to continue forwarding IP packets following an RP switchover.

Usually, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in what is called a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With NSF, peer networking devices do not experience routing flaps. Data traffic is forwarded through intelligent line cards or dual forwarding processors (FPs) while the standby RP assumes control from the failed active RP during a switchover. The ability of line cards and FPs to remain up through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active RP is key to NSF operation.

The NSF feature provides the following benefits:

- Improved network availability—NSF continues forwarding network traffic and application state information so that user session information is maintained after a switchover.
- Overall network stability—Network stability may be improved with the reduction in the number of route flaps that had been created when devices in the network failed and lost their routing tables.
- Neighboring devices do not detect link flapping—Because the interfaces remain up across a switchover, neighboring devices do not detect a link flap (that is, the link does not go down and come back up).
- Prevention of routing flaps—Because SSO continues forwarding network traffic in the event of a switchover, routing flaps are avoided.
- No loss of user sessions—User sessions established prior to the switchover are maintained.

NSF always runs together with SSO. SSO supported protocols and applications must be high-availability (HA)-aware. A feature or protocol is HA-aware if it maintains, either partially or completely, undisturbed operation during an RP switchover. For some HA-aware protocols and applications, state information is synchronized from the active to the standby processor.

EIGRP NSF Operations

Cisco NSF is supported by the EIGRP protocol for routing and by CEF for forwarding. EIGRP depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. Once the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF, in turn, updates the line cards with the new FIB information.

EIGRP nonstop forwarding (NSF) capabilities are exchanged by EIGRP peers in hello packets. The NSF-capable device notifies its neighbors that an NSF restart operation has started by setting the restart (RS) bit in a hello packet. When an NSF-aware device receives notification from an NSF-capable neighbor that an NSF-restart operation is in progress, the NSF-capable and NSF-aware devices immediately exchange their topology tables. The NSF-aware device sends an end-of-table (EOT) update packet when the transmission of its topology table is complete. The NSF-aware device then performs the following actions to assist the NSF-capable device:

- The EIGRP hello hold timer is expired to reduce the time interval set for hello packet generation and transmission. This allows the NSF-aware device to reply to the NSF-capable device more quickly reducing the amount of time required for the NSF-capable device to rediscover neighbors and rebuild the topology table.
- The route-hold timer is started. This timer is used to set the period of time that the NSF-aware device will hold known routes for the NSF-capable neighbor.
- The NSF-aware device notes in the peer list that the NSF-capable neighbor is restarting, maintains adjacency, and holds known routes for the NSF-capable neighbor until the neighbor signals that it is ready for the NSF-aware device to send its topology table or the route-hold timer expires. If the route-hold timer expires on the NSF-aware device, the NSF-aware device will discard held routes and treat the NSF-capable device as a new device joining the network and reestablishing adjacency accordingly.
- The NSF-aware device will continue to send queries to the NSF-capable device that is still converging after switchover, effectively extending the time before a stuck-in-active (SIA) condition can occur.

When the switchover operation is complete, the NSF-capable device notifies its neighbors that it has reconverged and has received all of their topology tables by sending an EOT update packet to the assisting devices. The NSF-capable device then returns to normal operation. The NSF-aware device will look for alternate paths (go

active) for any routes that are not refreshed by the NSF-capable (restarting device). The NSF-aware device will then return to normal operation. If all paths are refreshed by the NSF-capable device, the NSF-aware device will immediately return to normal operation.

NSF-aware devices are completely compatible with non-NSF-aware or non-NSF-capable neighbors in an EIGRP network. A non-NSF-aware neighbor will ignore NSF capabilities and reset adjacencies and otherwise maintain the peering sessions normally.

How to Configure EIGRP Nonstop Forwarding

Configuring and Verifying EIGRP NSF

Repeat this task on each peer device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **nsf**
5. **timers nsf converge** *seconds*
6. **timers nsf signal** *seconds*
7. **timers graceful-restart purge-time** *seconds*
8. **end**
9. **show ip protocols**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *as-number* **Example:** `Device(config)# router eigrp 109`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	**nsf** **Example:**	Enables NSF capabilities.

	Command or Action	Purpose
	`Device(config-router)# nsf`	• This command is enabled by default. To disable nonstop forwarding capability, use the **no** form of this command.
Step 5	**timers nsf converge** *seconds* **Example:** `Device(config-router)# timers nsf converge 120`	Use this optional command to adjust the maximum time that the restarting device will wait for the EOT notification from an NSF-capable or NSF-aware peer. • Enter this command on NSF-capable devices only.
Step 6	**timers nsf signal** *seconds* **Example:** `Device(config-router)# timers nsf signal 20`	Use this optional command to adjust the maximum time for the initial restart period. • Enter this command on NSF-capable devices only.
Step 7	**timers graceful-restart purge-time** *seconds* **Example:** `Device(config-router)# timers graceful-restart purge-time 240`	Use this optional command to set the route-hold timer to determine how long an NSF-aware EIGRP device will hold routes for an inactive peer.
Step 8	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 9	**show ip protocols** **Example:** `Device# show ip protocols`	Displays the parameters and current state of the active routing protocol process.

Troubleshooting EIGRP Nonstop Forwarding

Use the following commands in any order to troubleshoot issues with nonstop forwarding using the EIGRP protocol.

SUMMARY STEPS

1. **enable**
2. **debug eigrp nsf**
3. **debug ip eigrp notifications**
4. **show cef nsf**
5. **show cef state**
6. **show ip cef**
7. **show ip eigrp neighbors detail**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **debug eigrp nsf**

Example:

```
Device# debug eigrp nsf
```

Displays notifications and information about NSF events for an EIGRP routing process.

Step 3 **debug ip eigrp notifications**

Example:

```
Device# debug ip eigrp notifications
```

Displays information and notifications for an EIGRP routing process. This output includes NSF notifications and events.

Step 4 **show cef nsf**

Example:

```
Device# show cef nsf
```

Displays the current NSF state of CEF on both the active and standby RPs.

Step 5 **show cef state**

Example:

```
Device# show cef state
```

Displays the CEF state on a networking device.

Step 6 **show ip cef**

Example:

```
Device# show ip cef
```

Displays entries in the FIB that are unresolved or displays a FIB summary.

Step 7 **show ip eigrp neighbors detail**

Example:

```
Device# show ip eigrp neighbors detail
```

Displays detailed information about neighbors discovered by EIGRP.

Configuration Examples for EIGRP Nonstop Forwarding

Example: EIGRP NSF

The following sample output shows that EIGRP NSF support is present in the installed software image.

- "EIGRP NSF-aware route hold timer is . . ." is displayed in the output for either NSF-aware or NSF-capable devices, and the default or user-defined value for the route-hold timer is displayed.
- "EIGRP NSF enabled" or "EIGRP NSF disabled" appears in the output only when the NSF capability is supported by the device.

```
Device# show ip protocols

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 100
  EIGRP NSF-aware route hold timer is 240s
  EIGRP NSF enabled
     NSF signal timer is 20s
     NSF converge timer is 120s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.4.9.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 168: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 137

EIGRP IPv6 NSF/GR

The EIGRP IPv6 NSF/GR feature allows a Nonstop Forwarding (NSF)-aware device that is running Enhanced Interior Gateway Routing Protocol (EIGRP) to forward IPv6 packets while EIGRP restarts after recovering from a failure.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for EIGRP IPv6 NSF/GR

- EIGRP (Enhanced Interior Gateway Routing Protocol) IPv6 must be configured on devices. You need not specify the **network** *network-number* command in EIGRP named mode. By default, EIGRP IPv6 enables EIGRP on all interfaces configured with an IPv6 address.
- Cisco software that supports Nonstop Forwarding (NSF) awareness or NSF capabilities must be installed.
- A redundant facility must be configured to notify EIGRP during a switchover and to notify whether the restart is due to a switchover or a device reboot.
- An NSF-aware device must be up and completely converged with the network before it can assist an NSF-capable device in an NSF restart operation.

- All neighboring devices participating in EIGRP NSF must be NSF-capable or NSF-aware.

Restrictions for EIGRP IPv6 NSF/GR

- Nonstop Forwarding (NSF) is supported on platforms that support high-availability systems.
- An Enhanced Interior Gateway Routing Protocol (EIGRP) NSF-aware network does not allow two neighbors to perform an NSF restart operation at the same time. However, neighbors can re-establish peering sessions after the NSF restart operation is complete.
- NSF for IPv6 is supported only in EIGRP named mode configurations.

Information About EIGRP IPv6 NSF/GR

EIGRP IPv6 NSF/GR

The EIGRP IPv6 NSF/GR feature allows a Nonstop Forwarding (NSF)-aware device that is running Enhanced Interior Gateway Routing Protocol (EIGRP) to forward IPv6 packets along routes that are known to a device that is performing a switchover operation. EIGRP peers retain adjacencies and routes learned from a restarting peer (the device that is undergoing a switchover), and the EIGRP peers continue to forward IPv6 packets to the restarting peer. The high-availability systems on the device retain the forwarding table and continue to forward IPv6 packets until the control plane (EIGRP) has converged on the restarting device.

NSF allows forwarding of IPv6 packets while the device restarts after a failure. Graceful Restart (GR) allows topology databases to resynchronize while maintaining neighbor relationships and forwarding paths.

Note NSF supports IPv4 in EIGRP classic mode and named mode configurations. NSF supports IPv6 in named mode. For more information about the EIGRP IPv4 NSF feature, see the "EIGRP Nonstop Forwarding Awareness" module in the *IP Routing: EIGRP Configuration Guide*.

EIGRP IPv6 NSF Timers

The EIGRP IPv6 NSF/GR feature supports three types of timers: the signal timer, the converge timer, and the graceful-restart purge-time timer.

Configure the signal timer to adjust the maximum time of the initial restart period. The restarting device sends hello packets with the restart-signal (RS) bit set. If the restarting device has not learned about any neighbor or any Nonstop Forwarding (NSF)-aware neighbor or has not received all updates from neighbors when the timer expires, the Routing Information Base (RIB) is notified for convergence. The default value for the signal timer is 20 seconds. The **timers nsf signal** command is used to configure the signal timer.

Configure the converge timer to adjust the maximum time that a restarting device waits for the end-of-table (EOT) indications from all neighbors. The default value for the converge timer is 120 seconds. The **timers nsf converge** command is used to configure the converge timer.

Configure the graceful-restart purge-time timer to adjust the maximum waiting time to receive the convergent signal from a restarting device. The graceful-restart purge-time timer is used when the NSF-aware peer does not receive the EOT indication from the restarting neighbor. When the graceful-restart purge-time timer expires, the Enhanced Interior Gateway Routing Protocol (EIGRP) peer scans the topology table for stale routes from the restarting neighbor and changes the stale routes to active. This process allows EIGRP peers to find alternate routes instead of waiting during a long switchover operation. The default value for the graceful-restart purge-time timer is 240 seconds. The **timers graceful-restart purge-time** command is used to configure the graceful-restart purge-time timer.

How to Configure EIGRP IPv6 NSF/GR

Enabling EIGRP IPv6 NSF/GR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *name*
4. **address-family ipv6 autonomous-system** *number*
5. **nsf**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *name* **Example:** `Device(config)# router eigrp e1`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.
Step 4	**address-family ipv6 autonomous-system** *number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 1`	Enters address family configuration mode to configure an EIGRP IPv6 routing instance.
Step 5	**nsf** **Example:**	Enables Nonstop Forwarding (NSF) for the specific address family on the device.

	Command or Action	Purpose
	`Device(config-router-af)# nsf`	
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Modifying EIGRP IPv6 NSF Timers

Perform this task to modify EIGRP IPv6 NSF timers. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *name*
4. **address-family ipv6 autonomous-system** *number*
5. **timers nsf signal** *seconds*
6. **timers nsf converge** *seconds*
7. **timers graceful-restart purge-time** *seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *name* **Example:** `Device(config)# router eigrp e1`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.
Step 4	**address-family ipv6 autonomous-system** *number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 1`	Enters address family configuration mode to configure an EIGRP IPv6 routing instance.

	Command or Action	Purpose
Step 5	**timers nsf signal** *seconds* **Example:** `Device(config-router-af)# timers nsf signal 15`	Sets the initial restart period, in seconds, for the restarting device to send hello packets with the restart-signal (RS) bit set.
Step 6	**timers nsf converge** *seconds* **Example:** `Device(config-router-af)# timers nsf converge 60`	Sets the maximum time, in seconds, that the restarting device must wait for end-of-table (EOT) indications from all neighbors.
Step 7	**timers graceful-restart purge-time** *seconds* **Example:** `Device(config-router-af)# timers graceful-restart purge-time 150`	Sets the graceful-restart purge-time timer to determine the period, in seconds, for which a Nonstop Forwarding (NSF)-aware device that is running EIGRP must hold routes for an inactive peer.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Verifying the EIGRP IPv6 NSF/GR Configuration

SUMMARY STEPS

1. **enable**
2. **show ipv6 protocols**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 protocols** **Example:** `Device# show ipv6 protocols`	Displays parameters and the current state of the active IPv6 routing protocol process. • The output of this command can be used to verify the EIGRP IPv6 NSF/GR configuration.

Monitoring EIGRP IPv6 NSF/GR Events

SUMMARY STEPS

1. **enable**
2. **debug eigrp nsf**
3. **debug eigrp address-family ipv6 notifications**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug eigrp nsf** **Example:** `Device# debug eigrp nsf`	Displays debugging information about NSF events on the console of the router.
Step 3	**debug eigrp address-family ipv6 notifications** **Example:** `Device# debug eigrp address-family ipv6 notifications`	Displays debugging information about Enhanced Interior Gateway Routing Protocol (EIGRP) address family IPv6 event notifications.

Configuration Examples for EIGRP IPv6 NSF/GR

Example: Configuring an EIGRP NSF Converge Timer

The following example shows how to adjust the maximum time that the restarting router waits for end-of-table (EOT) indications from all neighbors:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 autonomous system 1
Device(config-router-af)# timers nsf converge 60
Device(config-router-af)# end
```

Example: Verifying the Configuration of EIGRP IPv6 NSF/GR on an NSF-Aware Device

The following is a sample output from the **show ipv6 protocols** command, which shows that EIGRP NSF is enabled, the graceful-restart purge-time timer is set to 260 seconds, the signal timer is set to 15 seconds, and the converge timer is set to 65 seconds:

```
Device> enable
Device# show ipv6 protocols

IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "bgp 1"
  IGP synchronization is disabled
  Redistribution:
    None
IPv6 Routing Protocol is "bgp multicast"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 1"
EIGRP-IPv6 VR(name) Address-Family Protocol for AS(1)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0 K6=0
  Metric rib-scale 128
  Metric version 64bit
  NSF-aware route hold timer is 260
  EIGRP NSF enabled
     NSF signal timer is 15s
     NSF converge timer is 65s
  Router-ID: 10.1.1.1
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 16
    Maximum hopcount 100
    Maximum metric variance 1
    Total Prefix Count: 0
    Total Redist Count: 0

  Interfaces:
  Redistribution:
    None
```

Additional References for EIGRP IPv6 NSF/GR

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco Express Forwarding (formerly known as CEF) commands	Cisco IOS IP Switching Command Reference
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference

Related Topic	Document Title
Nonstop Forwarding (NSF)	• "Cisco Nonstop Forwarding" module in the Stateful Switchover Deployment Guide • "Cisco Nonstop Forwarding" module in the *High Availability Configuration Guide* • "EIGRP Nonstop Forwarding Awareness" module in the *IP Routing: EIGRP Configuration Guide*
Command Lookup Tool	http://tools.cisco.com/Support/CLILookup

Standards and RFCs

Standard/RFC	Title
RFC 4724	*Graceful Restart Mechanism for BGP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 169: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 138

EIGRP Prefix Limit Support

The EIGRP Prefix Limit Support feature introduces the capability to limit the number of prefixes per VPN routing/forwarding instance (VRF) that are accepted from a specific peer or to limit all prefixes that are accepted by an Enhanced Interior Gateway Routing Protocol (EIGRP) process through peering and redistribution. This feature is designed to protect the local router from external misconfiguration that can negatively impact local system resources; for example, a peer that is misconfigured to redistribute full Border Gateway Protocol (BGP) routing tables into EIGRP. This feature is enabled under the IPv4 VRF address family and can be configured to support the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature.

For more information about EIGRP MPLS VPN configuration, refer to the EIGRP MPLS VPN PE-CE Site of Origin module.

Prerequisites for EIGRP Prefix Limit Support

- Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) services have been configured between the Provider Edge (PE) routers and the customer edge (CE) routers at the customer sites.

Restrictions for EIGRP Prefix Limit Support

- This feature is supported only under the IPv4 VRF address family and can be used only to limit the number of prefixes that are accepted through a VRF.
- The EIGRP Prefix Limiting Support feature is enabled only under the IPv4 VRF address-family. A peer that is configured to send too many prefixes or a peer that rapidly advertises and then withdraws prefixes can cause instability in the network. This feature can be configured to automatically reestablish a disabled peering session at the default or user-defined time interval or when the maximum-prefix limit is not exceeded. However, the configuration of this feature alone cannot change or correct a peer that is sending

an excessive number of prefixes. If the maximum-prefix limit is exceeded, you will need to reconfigure the maximum-prefix limit or reduce the number of prefixes that are sent from the peer.

Information About EIGRP Prefix Limit Support

Misconfigured VPN Peers

In MPLS VPNs, the number of routes that are permitted in the VRF is configured with the **maximum routes** VRF configuration command. However, limiting the number routes permitted in the VPN does not protect the local router from a misconfigured peer that sends an excessive number of routes or prefixes. This type of external misconfiguration can have a negative effect on the local router by consuming all available system resources (CPU and memory) in processing prefix updates. This type of misconfiguration can occur on a peer that is not within the control of the local administrator.

EIGRP Prefix Limit Support Overview

The EIGRP Prefix Limit Support feature provides the ability to configure a limit on the number of prefixes that are accepted from EIGRP peers or learned through redistribution. This feature can be configured on per-peer or per-process basis and can be configured for all peers and processes. This feature is designed to protect the local router from misconfigured external peers by limiting the amount of system resources that can be consumed to process prefix updates.

External Peer Router Protection

This feature can be configured to protect an individual peering session or protect all peering sessions. When this feature is enabled and the maximum-prefix limit has been exceeded, the router will tear down the peering session, clear all routes that were learned from the peer, and then place the peer in a penalty state for the default or user-defined time period. After the penalty time period expires, normal peering will be reestablished.

Redistributed Prefix Number Limiting

This feature can be configured to limit the number of prefixes that are accepted into the EIGRP topology table through redistribution from the Routing Information Base (RIB). All sources of redistribution are processed cumulatively. When the maximum-prefix limit is exceeded, all routes learned through redistribution are discarded and redistribution is suspended for the default or user-defined time period. After the penalty time period expires, normal redistribution will occur.

EIGRP Process Level Router Protection

This feature can be configured to protect the router at the EIGRP process level. When this feature is configured at the EIGRP process level, the maximum-prefix limit is applied to all peering sessions and to route redistribution. When the maximum-prefix limit is exceeded, all sessions with the remote peers are torn down, all routes learned from remote peers are removed from the topology and routing tables, all routes learned through redistribution are discarded, and redistribution and peering are suspended for the default or user-defined time period.

EIGRP Prefix Limiting Warning-Only Mode

The EIGRP Prefix Limit Support feature has two modes of operation. This feature can control peering and redistribution per default and user-defined values or this feature can operate in warning-only mode. In warning-only mode the router will monitor the number of prefixes learned through peering and/or redistribution but will not take any action when the maximum-prefix limit is exceeded. Warning-only mode is activated only when the **warning-only** keyword is configured for any of the maximum-prefix limit commands. Only syslog messages are generated when this mode of operation is enabled. Syslog messages can be sent to a syslog server or printed in the console. These messages can be buffered or rate limited per standard Cisco IOS XE system logging configuration options.

EIGRP Prefix Limiting Restart Reset and Dampening Timers and Counters

The EIGRP Prefix Limit Support feature provides two user-configurable timers, a restart counter, and a dampening mechanism. When the maximum-prefix limit is exceeded, peering and/or redistribution is suspended for a default or user-defined time period. If the maximum-prefix limit is exceeded too often, redistribution and/or peering will be suspended until manual intervention is taken.

Restart Timer

The restart timer determines how long the router will wait to form an adjacency or accept redistributed routes from the RIB after the maximum-prefix limit has been exceeded. The default restart-time period is 5 minutes.

Restart Counter

The restart counter determines the number of times a peering session can be automatically reestablished after the peering session has been torn down or after the redistributed routes have been cleared and relearned because the maximum-prefix limit has been exceeded. The default restart-count limit is three.

Caution After the restart count limit has been crossed, you will need to enter the **clear ip route ***, **clear ip eigrp neighbor**, or **clear eigrp address-family neighbor**command to restore normal peering and redistribution.

Reset Timer

The reset timer is used to configure the router to reset the restart count to 0 after the default or configured reset-time period has expired. This timer is designed to provide an administrator with control over long-and medium-term accumulated penalties. The default reset-time period is 15 minutes.

Dampening Mechanism

The dampening mechanism is used to apply an exponential decay penalty to the restart-time period each time the maximum-prefix limit is exceeded. The half-life for the decay penalty is 150 percent of the default or user-defined restart-time value in minutes. This mechanism is designed to identify and suppress unstable peers. It is disabled by default.

How to Configure the Maximum-Prefix Limit

Note From Cisco IOS XE 17.13.1a, if the EIGRP process enters into a suspended (pending or down) state the router will no longer establish neighborships with new peers and thus cease to transmit and stop processing hello packets. For more information see, Cisco IOS IP Routing: EIGRP Command Reference

Configuring the Maximum Number of Prefixes Accepted from Peering Sessions Autonomous System Configuration

The maximum-prefix limit can be configured for all peering sessions or individual peering sessions with the **neighbor maximum-prefix**(EIGRP) command. When the maximum-prefix limit is exceeded, the session with the remote peer is torn down and all routes learned from the remote peer are removed from the topology and routing tables. The maximum-prefix limit that can be configured is limited only by the available system resources on the router.

Note In EIGRP, **neighbor** commands have been used traditionally to configure static neighbors. In the context of this feature, however, the **neighbor maximum-prefix** command can be used to configure the maximum-prefix limit for both statically configured and dynamically discovered neighbors.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

Before you begin

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note
- This task can be configured only in IPv4 VRF address family configuration mode.
- When you configure the **neighbor maximum-prefix** command to protect a single peering session, only the maximum-prefix limit, the percentage threshold, the warning-only configuration options can be configured. Session dampening, restart, and reset timers are configured on a global basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*

4. **address-family ipv4** [**unicast**][**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **description** *text*
6. **neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*] [**warning-only**]
7. **neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**unicast**][**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf vrf1 autonomous-system 4453`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **description** *text* **Example:** `Router(config-router-af)# neighbor 172.16.2.3 description peer with example.com`	(Optional) Associates a description with a neighbor.
Step 6	**neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*] [**warning-only**] **Example:** `Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 10000 80 warning-only`	Limits the number of prefixes that are accepted from the specified EIGRP neighbor.
Step 7	**neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]	Limits the number of prefixes that are accepted from all EIGRP neighbors.

	Command or Action	Purpose
	Example: `Router(config-router-af)# neighbor maximum-prefix 10000 80 warning-only`	
Step 8	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

If an individual peer or all peers have exceeded the maximum-prefix limit the same number of times as the default or user-defined restart-count value, the individual session or all sessions will need to be manually reset with the **clear ip route*** or **clear ip eigrp neighbor** command before normal peering can be reestablished.

Configuring the Maximum Number of Prefixes Accepted from Peering Sessions Named Configuration

The maximum-prefix limit can be configured for all peering sessions or individual peering sessions with the **neighbor maximum-prefix** (EIGRP) command. When the maximum-prefix limit is exceeded, the session with the remote peer is torn down and all routes learned from the remote peer are removed from the topology and routing tables. The maximum-prefix limit that can be configured is limited only by the available system resources on the router.

Note In EIGRP, **neighbor** commands have been used traditionally to configure static neighbors. In the context of this feature, however, the **neighbor maximum-prefix** command can be used to configure the maximum-prefix limit for both statically configured and dynamically discovered neighbors.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

Before you begin

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note

- This task can be configured only in IPv4 VRF address family configuration mode.
- When you configure the **neighbor maximum-prefix** command to protect a single peering session, only the maximum-prefix limit, the percentage threshold, and the warning-only configuration options can be configured. Session dampening, restart, and reset timers are configured on a global basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **description** *text*
6. **neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*] [**warning-only**]
7. **neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
8. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **description** *text* **Example:**	(Optional) Associates a description with a neighbor.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor 172.16.2.3 description peer with example.com`	
Step 6	**neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*] [**warning-only**] **Example:** `Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 10000 80 warning-only`	Limits the number of prefixes that are accepted from the specified EIGRP neighbor.
Step 7	**neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af)# neighbor maximum-prefix 10000 80 warning-only`	Limits the number of prefixes that are accepted from all EIGRP neighbors.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.

Troubleshooting Tips

If an individual peer or all peers have exceeded the maximum-prefix limit the same number of times as the default or user-defined restart-count value, the individual session or all sessions will need to be manually reset with the **clear ip route*** or **clear eigrp address-family neighbors** command before normal peering can be reestablished.

Configuring the Maximum Number of Prefixes Learned Through Redistribution Autonomous System Configuration

The maximum-prefix limit can be configured for prefixes learned through redistribution with the **redistribute maximum-prefix** (EIGRP) command. When the maximum-prefix limit is exceeded, all routes learned from the RIB will be discarded and redistribution will be suspended for the default or user-defined time period. The maximum-prefix limit that can be configured for redistributed prefixes is limited only by the available system resources on the router.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

Before you begin

- VRFs have been created and configured.

- EIGRP peering is established through the MPLS VPN.

Note This task can be configured only in IPv4 VRF address family configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
5. **redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf VRF1`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*]	**warning-only**] **Example:** `Router(config-router-af)# redistribute maximum-prefix 10000 80 reset-time 10 restart 2`	Limits the number of prefixes redistributed into an EIGRP process.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear ip eigrp neighbors** command will need to be entered before normal redistribution will occur.

Configuring the Maximum Number of Prefixes Learned Through Redistribution Named Configuration

The maximum-prefix limit can be configured for prefixes learned through redistribution with the **redistribute maximum-prefix**(EIGRP) command. When the maximum-prefix limit is exceeded, all routes learned from the RIB will be discarded and redistribution will be suspended for the default or user-defined time period. The maximum-prefix limit that can be configured for redistributed prefixes is limited only by the available system resources on the router.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

Before you begin

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note This task can be configured only in IPv4 VRF address family topology configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **topology base**
7. **redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]

8. **exit-af-topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Router(config-router-af)# network 172.16.0.0`	Specifies the network for an EIGRP address family routing process.
Step 6	**topology base** **Example:** `Router(config-router-af)# topology base`	Configures an EIGRP process to route traffic under the specified topology instance and enters address family topology configuration mode.
Step 7	**redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af-topology)# redistribute maximum-prefix 10000 80 reset-time 10 restart 2`	Limits the number of prefixes redistributed into an EIGRP process.
Step 8	**exit-af-topology** **Example:**	Exits address family topology configuration mode.

	Command or Action	Purpose
	`Router(config-router-af-topology)# exit-af-topology`	

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear eigrp address-family neighbors**command will need to be entered before normal redistribution will occur.

Configuring the Maximum-Prefix Limit for an EIGRP Process Autonomous System Configuration

The maximum-prefix limit can be configured for an EIGRP process to limit the number prefixes that are accepted from all sources. This task is configured with the **maximum-prefix**command. When the maximum-prefix limit is exceeded, sessions with the remote peers are brought down and all routes learned from remote peers are removed from the topology and routing tables. Also, all routes learned from the RIB are discarded and redistribution is suspended for the default or user-defined time period.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

Before you begin

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note This task can be configured only in IPv4 VRF address family configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **address-family ipv4** [**unicast**] **vrf** *vrf-name*[**autonomous-system** *autonomous-system-number*]
5. **maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**unicast**] **vrf** *vrf-name*[**autonomous-system** *autonomous-system-number*] **Example:** `Router(config-router)# address-family ipv4 vrf VRF1`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af)# maximum-prefix 10000 80 reset-time 10 restart 2`	Limits the number of prefixes that are accepted under an address family by an EIGRP process. • The example configures a maximum-prefix limit of 10,000 prefixes, a reset time period of 10 minutes, a warning message to be displayed at 80 percent of the maximum-prefix limit, and a restart time period of 2 minutes.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear ip eigrp neighbors** command will need to be entered before normal redistribution will occur.

Configuring the Maximum-Prefix Limit for an EIGRP Process Named Configuration

The maximum-prefix limit can be configured for an EIGRP process to limit the number prefixes that are accepted from all sources. This task is configured with the **maximum-prefix**command. When the maximum-prefix limit is exceeded, sessions with the remote peers are brought down and all routes learned

from remote peers are removed from the topology and routing tables. Also, all routes learned from the RIB are discarded and redistribution is suspended for the default or user-defined time period.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

Before you begin

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note This task can be configured only in IPv4 VRF address family topology configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
6. **exit-address-family**
7. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Creates an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.

	Command or Action	Purpose
Step 4	**address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af)# maximum- prefix 10000 80 reset-time 10 restart 2 warning-only`	Limits the number of prefixes that are accepted under an address family by an EIGRP process. • The example configures a maximum-prefix limit of 10,000 prefixes, a reset time period of 10 minutes, a warning message to be displayed at 80 percent of the maximum-prefix limit, and a restart time period of 2 minutes.
Step 6	**exit-address-family** **Example:** `Router(config-router-af)# exit-af-topology`	Exits address family configuration mode.
Step 7	**show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting** **Example:** `Router# show eigrp address-family ipv4 22 accounting`	(Optional) Displays prefix accounting information for EIGRP processes. **Note** Connected and summary routes are not listed individually in the output from this **show** command but are counted in the total aggregate count per process.

Example

The following is sample output from the **show eigrp address-family accounting** command:

```
Router# show eigrp address-family ipv4 22 accounting
EIGRP-IPv4 VR(saf) Accounting for AS(22)/ID(10.0.0.1)
Total Prefix Count: 3  States: A-Adjacency, P-Pending, D-Down
State Address/Source    Interface       Prefix   Restart  Restart/
                                        Count     Count   Reset(s)
 A    10.0.0.2          Et0/0               2         0        0
 P    10.0.2.4          Se2/0               0         2        114
 D    10.0.1.3          Et0/0               0         3        0
```

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear eigrp address-family neighbors**command will need to be entered before normal redistribution will occur.

Configuration Examples for Configuring the Maximum-Prefix Limit

Example Configuring the Maximum-Prefix Limit for a Single Peer--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for a single peer. The maximum limit is set to 1000 prefixes, and the warning threshold is set to 80 percent. When the maximum-prefix limit is exceeded, the session with this peer will be torn down, all routes learned from this peer will be removed from the topology and routing tables, and this peer will be placed in a penalty state for 5 minutes (default penalty value).

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf VRF1
Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 1000 80
Router(config-router-af)# end
```

Note If the max prefix limit at process level and neighbor level is set together then the max prefix limit at process level will take precedence. And when the max prefix limit at neighbor level is set greater than the max prefix limit set at process level, the device displays this message:

```
Max prefix limit at neighbor level is set to a value (%d)
greater than max prefix limit at process level (%d)
```

Example Configuring the Maximum-Prefix Limit for a Single Peer--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for a single peer. The maximum limit is set to 1000 prefixes, and the warning threshold is set to 80 percent. When the maximum-prefix limit is exceeded, the session with this peer will be torn down, all routes learned from this peer will be removed from the topology and routing tables, and this peer will be placed in a penalty state for 5 minutes (default penalty value).

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 1000 80
Router(config-router-af)# exit-address-family
```

Note If the maximum prefix limit at process level and neighbor level is set together then the max prefix limit at process level will take precedence. And when the max prefix limit at neighbor level is set greater than the max prefix limit set at process level, the device displays this message:

```
Max prefix limit at neighbor level is set to a value (%d)
greater than max prefix limit at process level (%d)
```

Example Configuring the Maximum-Prefix Limit for All Peers--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for all peers. The maximum limit is set to 10,000 prefixes, the warning threshold is set to 90 percent, the restart timer is set to 4 minutes, a decay penalty is configured for the restart timer with the **dampened** keyword, and all timers are configured to be reset to 0 every 60 minutes. When the maximum-prefix limit is exceeded, all peering sessions will be torn down, all routes learned from all peers will be removed from the topology and routing tables, and all peers will be placed in a penalty state for 4 minutes (user-defined penalty value). A dampening exponential decay penalty will also be applied.

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf VRF1
Router(config-router-af)# neighbor maximum-prefix 1500 90 dampened reset-time 60 restart 4
Router(config-router-af)# end
```

Note If the maximum prefix limit at process level and neighbor level is set together then the max prefix limit at process level will take precedence. And when the max prefix limit at neighbor level is set greater than the max prefix limit set at process level, the device displays this message:

```
Max prefix limit at neighbor level is set to a value (%d)
greater than max prefix limit at process level (%d)
```

Example Configuring the Maximum-Prefix Limit for All Peers--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for all peers. The maximum limit is set to 10,000 prefixes, the warning threshold is set to 90 percent, the restart timer is set to 4 minutes, a decay penalty is configured for the restart timer with the **dampened** keyword, and all timers are configured to be reset to 0 every 60 minutes. When the maximum-prefix limit is exceeded, all peering sessions will be torn down, all routes learned from all peers will be removed from the topology and routing tables, and all peers will be placed in a penalty state for 4 minutes (user-defined penalty value). A dampening exponential decay penalty will also be applied.

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# neighbor maximum-prefix 10000 90 dampened reset-time 60 restart
4
Router(config-router-af)# exit-address-family
```

Note If the maximum prefix limit at process level and neighbor level is set together then the max prefix limit at process level will take precedence. And when the max prefix limit at neighbor level is set greater than the max prefix limit set at process level, the device displays this message:

```
Max prefix limit at neighbor level is set to a value (%d)
greater than max prefix limit at process level (%d)
```

Example Configuring the Maximum-Prefix Limit for Redistributed Routes--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for routes learned through redistribution. The maximum limit is set to 5000 prefixes and the warning threshold is set to 95 percent. When the number of prefixes learned through redistribution reaches 4750 (95 percent of 5000), warning messages will be displayed in the console. Because the **warning-only** keyword is configured, the topology and routing tables will not be cleared and route redistribution will not be placed in a penalty state.

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf VRF1
Router(config-router-af)# redistribute maximum-prefix 5000 95 warning-only
Router(config-router-af)# end
```

Note When the maximum prefix limit is configured at both the process level and redistribution level, the limit set at the process level will take precedence. In cases where the max prefix limit at redistribution level is set greater than the max prefix limit set at process level, the device displays this message:

```
Max prefix limit at redistribution level is set to a value (%d)
greater than max prefix limit at process level (%d)
```

Example Configuring the Maximum-Prefix Limit for Redistributed Routes--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for routes learned through redistribution. The maximum limit is set to 5000 prefixes and the warning threshold is set to 95 percent. When the number of prefixes learned through redistribution reaches 4750 (95 percent of 5000), warning messages will be displayed in the console. Because the **warning-only** keyword is configured, the topology and routing tables will not be cleared and route redistribution will not be placed in a penalty state.

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# network 172.16.0.0
Router(config-router-af)# topology base
Router(config-router-af-topology)# redistribute maximum-prefix 5000 95 warning-only
Router(config-router-af-topology)# exit-af-topology
```

Note When the maximum prefix limit is configured at both the process level and redistribution level, the limit set at the process level will take precedence. In cases where the max prefix limit at redistribution level is set greater than the max prefix limit set at process level, the device displays this message:

```
Max prefix limit at redistribution level is set to a value (%d)
greater than max prefix limit at process level (%d)
```

Example Configuring the Maximum-Prefix Limit for an EIGRP Process--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for an EIGRP process, which includes routes learned through redistribution and routes learned through EIGRP peering sessions. The maximum limit is set to 50,000 prefixes. When the number of prefixes learned through redistribution reaches 37,500 (75 percent of 50,000), warning messages will be displayed in the console.

When the maximum-prefix limit is exceeded, all peering sessions will be reset, the topology and routing tables will be cleared, and redistributed routes and all peering sessions will be placed in a penalty state.

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf RED
Router(config-router-af)# maximum-prefix 50000
Router(config-router-af)# end

\
```

Note

- When the max prefix limit at process level is set lower than the max prefix limit set at redistribute level, the device displays this message:

  ```
  Max prefix limit at process level is set to a value (%d)lower than max prefix limit
  at redistribute level (%d)
  ```

- When the max prefix limit at process level is set lower than the max prefix limit set at neighbor level, the device displays this message:

  ```
  (%d) lower than max prefix limit at neighbor level (%d)
  ```

Example Configuring the Maximum-Prefix Limit for an EIGRP Process--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for an EIGRP process, which includes routes learned through redistribution and routes learned through EIGRP peering sessions. The maximum limit is set to 50,000 prefixes. When the number of prefixes learned through redistribution reaches 37,500 (75 percent of 50,000), warning messages will be displayed in the console.

When the maximum-prefix limit is exceeded, all peering sessions will be reset, the topology and routing tables will be cleared, and redistributed routes and all peering sessions will be placed in a penalty state.

```
Router(config)# router eigrp virtual-name
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# maximum-prefix 50000
Router(config-router-af)# exit-address-family
```

Note

- When the max prefix limit at process level is set lower than the max prefix limit set at redistribute level, the device displays this message:

```
Max prefix limit at process level is set to a value (%d) lower than max prefix limit
at redistribute level (%d)
```

- When the max prefix limit at process level is set lower than the max prefix limit set at neighbor level, the device displays this message:

```
(%d) lower than max prefix limit at neighbor level (%d)
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
EIGRP autonomous system configuration and EIGRP named configuration	Configuring EIGRP module
BGP cost community configuration tasks for EIGRP MPLS VPN PE-CE	BGP Cost Community module of the *Cisco IOS IP Routing: BGP Configuration Guide*
Basic EIGRP configuration tasks	Configuring EIGRP module
EIGRP MPLS VPN configuration tasks	EIGRP MPLS VPN PE-CE Site of Origin (SoO) module
MPLS VPNs configuration tasks	Configuring MPLS Layer 3 VPNs module of the *Cisco IOS Multiprotocol Label Switching Configuration Guide*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 170: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 139

EIGRP Support for Route Map Filtering

The EIGRP Support for Route Map Filtering feature enables Enhanced Interior Gateway Routing Protocol (EIGRP) to interoperate with other protocols to leverage additional routing functionality by filtering inbound and outbound traffic based on complex route map options. Several extended filtering options are introduced to provide EIGRP-specific match choices.

Information About EIGRP Support for Route Map Filtering

EIGRP Route Map Support

EIGRP support for route map filtering enables EIGRP to interoperate with other protocols by filtering inbound and outbound traffic based on route map options. Additional EIGRP-specific match choices are available to allow flexibility in fine-tuning EIGRP network operations.

EIGRP supports the route map filtering capability that exists for other routing protocols to filter routes being redistributed into their protocol. For more details about understanding and configuring route maps, see the Enabling Policy Routing section of the Configuring IP Routing Protocol-Independent Features module of the *Cisco IOS XE IP Routing: Protocol-Independent Configuration Guide*, Release 2.

Match options allow EIGRP to filter internal and external routes based on source protocols, to match a metric against a range, and to match on an external protocol metric.

EIGRP can be configured to filter traffic using a route map and the **redistribute** or **distribute-list** command. Using a route map with the **redistribute** command allows routes that are redistributed from the routing table to be filtered with a route map before being admitted into an EIGRP topology table. Routes that are dynamically received from, or advertised to, EIGRP peers can be filtered by adding a route map option to the **distribute-list** command.

A route map may be configured with both the **redistribute** and the **distribute-list** commands in the same routing process. When a route map is used with a **distribute-list** command that is configured for inbound or outbound filtering, route packets that are learned from or advertised to EIGRP peers can be processed with the route map to provide better control of route selection during the route exchange process. Redistribution

serves as a mechanism to import routes into the EIGRP topology table from a routing table. A route map configured with the **redistribute** command adds flexibility to the redistribution capability and results in a more specific redistributed route selection.

The use of route maps to filter traffic is the same for both autonomous-system configurations and named configurations. See the Configuring EIGRP module for more information about autonomous system and named configurations.

Demands for EIGRP to interoperate with other protocols and flexibility in fine-tuning network operation necessitate the capability to filter traffic using a route map.

Note The **set metric +/-** command, which specifies the relative change of metric, is not supported with EIGRP redistribution route-maps. If configured, it is interpreted as the **set metric** command with the sign omitted, and can cause unexpected behavior in the configuration. It is recommended to not use the **set metric +/-** command with EIGRP redistribution route-map configuration.

How to Configure EIGRP Support for Route Map Filtering

Setting EIGRP Tags Using a Route Map for Autonomous System Configurations

Perform this task to set EIGRP tags for autonomous system configurations using a route map. The EIGRP metrics used for filtering are configured within a route map. The first match clause defines EIGRP routes that contain an external protocol metric between 400 and 600 inclusive; the second match clause defines EIGRP external routes that match a source protocol of BGP and the autonomous system 45000. When the two match clauses are true, a tag value of the destination routing protocol is set to 5. This route map can be used with the **distribute-list** command, see the Example Setting EIGRP Tags Using a Route Map--Autonomous System Configuration Examples, on page 1880 for an example configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match metric** {*metric-value*| **external** *metric-value*} [**+-** *deviation-number*]
5. **match source-protocol** *source-protocol* [*autonomous-system-number*]
6. **set tag** *tag-value*
7. **exit**
8. **router eigrp** *as-number*
9. **network** *ip-address*
10. **distribute-list route-map** *map-tag* **in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map metric-range`	Enters route-map configuration mode.
Step 4	**match metric** {*metric-value*| **external** *metric-value*} [**+-** *deviation-number*] **Example:** `Router(config-route-map)# match metric external 500 +- 100`	Specifies a match clause that filters inbound updates that match an internal or external protocol metric. • *metric-value* --Internal protocol metric, which can be an EIGRP five-part metric. The range is from 1 to 4294967295. • **external** --External protocol metric. The range is from 1 to 4294967295. • **+-** *deviation-number* --(Optional) Represents a standard deviation. The deviation can be any number. There is no default. **Note** When you specify a metric deviation with the + and - keywords, the router will match any metric that falls inclusively in that range. **Note** The external protocol metric is not the same as the EIGRP assigned route metric, which is a figure computed from EIGRP vectorized metric components (delay, bandwidth, reliability, load, and MTU).
Step 5	**match source-protocol** *source-protocol* [*autonomous-system-number*] **Example:** `Router(config-route-map)# match source-protocol bgp 45000`	Specifies a match clause that matches external routes from sources that match the source protocol. • *source-protocol* --Protocol to match. The valid keywords are **bgp**, **connected**, **eigrp**, **isis**, **ospf**, **rip**, and **static**. There is no default. • *autonomous-system-number* --(Optional) Autonomous system number. The *autonomous-system-number* argument is not applicable to the **connected**, **static**, and **rip** keywords. The range is from 1 to 65535. There is no default.

	Command or Action	Purpose
Step 6	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 5`	Sets a tag value on the route in the destination routing protocol when all the match criteria of a route map are met.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 8	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Configures the EIGRP routing process and enters router configuration mode.
Step 9	**network** *ip-address* **Example:** `Router(config-router)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 10	**distribute-list route-map** *map-tag* **in** **Example:** `Router(config-router)# distribute-list route-map metric-range in`	Filters networks received in updates.

Setting EIGRP Tags Using a Route Map for Named Configurations

Perform this task to set EIGRP tags for named configurations using a route map. The EIGRP metrics used for filtering are configured within a route map. The first match clause defines EIGRP routes that contain an external protocol metric between 400 and 600 inclusive; the second match clause defines EIGRP external routes that match a source protocol of BGP and the autonomous system 45000. When the two match clauses are true, a tag value of the destination routing protocol is set to 5. This route map can be used with the **distribute-list** command, see the #unique_2114 unique_2114_Connect_42_ GUID-AE466629-2BD8-4ACB-818D-2A916B4BDA4F, on page 1880for an example configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **set metric** *bandwidth delay reliability loading mtu*
5. **match ip route-source** {*access-list-number*| *access-list-name*} [*...access-list-number* | *...access-list-name*]
6. **match metric** {*metric-value*| **external** *metric-value*} [**+-** *deviation-number*]
7. **match source-protocol** *source-protocol* [*autonomous-system-number*]
8. **set tag** *tag-value*

9. **exit**
10. **router eigrp** *virtual-instance-name*
11. Do one of the following:
    - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
    - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
12. **network** *ip-address* [*wildcard-mask*]
13. **af-interface** {**default** | *interface-type interface-number*}
14. **next-hop-self**
15. **exit-af-interface**
16. **topology** {**base** | *topology-name* **tid** *number*}
17. **distribute-list route-map** *map-tag* **in**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map metric-range`	Enters route-map configuration mode.
Step 4	**set metric** *bandwidth delay reliability loading mtu* **Example:** `Router(config-route-map)# set metric 10000 10 255 1 1500`	(Optional) Sets the metric value for EIGRP in a route map.
Step 5	**match ip route-source** {*access-list-number*| *access-list-name*} [*...access-list-number* | *...access-list-name*] **Example:** `Router(config-route-map)# match ip route-source 5 80`	Redistributes routes that have been advertised by routers and access servers at the address specified by the access lists.
Step 6	**match metric** {*metric-value*| **external** *metric-value*} [**+-** *deviation-number*]	Specifies a match clause that includes EIGRP routes that match an internal or external protocol metric.

	Command or Action	Purpose
	Example: `Router(config-route-map)# match metric external 500 +- 100`	• *metric-value* --Internal protocol metric, which can be an EIGRP five-part metric. The range is from 1 to 4294967295. • **external** --External protocol metric. The range is from 1 to 4294967295. • **+-** *deviation-number* --(Optional) Represents a standard deviation. The deviation can be any number. There is no default. **Note** When you specify a metric deviation with the **+** and **-** keywords, the router will match any metric that falls inclusively in that range. **Note** The external protocol metric is not the same as the EIGRP assigned route metric, which is a figure computed from EIGRP vectorized metric components (delay, bandwidth, reliability, load, and MTU).
Step 7	**match source-protocol** *source-protocol* [*autonomous-system-number*] **Example:** `Router(config-route-map)# match source-protocol bgp 45000`	Specifies a match clause that includes EIGRP external routes that match a source protocol. • *source-protocol* --Protocol to match. The valid keywords are **bgp**, **connected**, **eigrp**, **isis**, **ospf**, **rip**, and **static**. There is no default. • *autonomous-system-number* --(Optional) Autonomous system number. The *autonomous-system-number* argument is not applicable to the **connected**, **static**, and **rip** keywords. The range is from 1 to 65535. There is no default.
Step 8	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 5`	Sets a tag value on the route in the destination routing protocol when all the match criteria of a route map are met.
Step 9	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 10	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Configures the EIGRP routing process and enters router configuration mode.

	Command or Action	Purpose
Step 11	Do one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 autonomous-system 45000`	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.
Step 12	**network** *ip-address* [*wildcard-mask*] **Example:** `Router(config-router-af)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 13	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Router(config-router-af)# af-interface default`	Enters address family interface configuration mode to configure interface-specific EIGRP commands.
Step 14	**next-hop-self** **Example:** `Router(config-router-af-interface)# next-hop-self`	Enables EIGRP to advertise routes with the local outbound interface address as the next hop.
Step 15	**exit-af-interface** **Example:** `Router(config-router-af-interface)# exit-af-interface`	Exits address-family interface configuration mode.
Step 16	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 17	**distribute-list route-map** *map-tag* **in** **Example:** `Router(config-router-af-topology)# distribute-list route-map metric-range in`	Filters networks received in updates.

Configuring EIGRP Route-map for Distribute-list in IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **topology** {**base** | *topology-name* **tid** *number*}
6. **distribute-list route-map** *map-tag* **in**
7. **distribute-list route-map** *map-tag* **out**
8. **exit-af-toplogy**
9. **exit-address-family**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
12. **set tag** *tag-value*
13. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
14. **match interface** *interface-type interface-number* [*...interface-type interface-number*]
15. **set tag** *tag-value*
16. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
17. **match metric** *bandwidth delay reliability loading mtu*
18. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
19. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
20. **set tag** *tag-value*
21. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
22. **match interface** *interface-type interface-number* [*...interface-type interface-number*]
23. **set tag** *tag-value*
24. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
25. **match metric** *bandwidth delay reliability loading mtu*
26. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual1`	Configures the EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv6 autonomous-system 1`	Enters address family configuration mode to configure an EIGRP IPv6 routing instance.
Step 5	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 6	**distribute-list route-map** *map-tag* **in** **Example:** `Router(config-router-af-topology)# distribute-list route-map map_in in`	Enables filtering of the networks received in EIGRP updates.
Step 7	**distribute-list route-map** *map-tag* **out** **Example:** `Router(config-router-af-topology)# distribute-list route-map map_out out`	Enables suppressing of networks from being advertised in the EIGRP updates.
Step 8	**exit-af-toplogy** **Example:** `Router(config-router-af-topology)# exit-af-topology`	Exits address-family topology configuration mode.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address-family configuration mode.
Step 10	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map map1 permit 10`	Enters route-map configuration mode. • Specifies route map name and set action to redistribute the route if the match criteria are met.
Step 11	**match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*} **Example:**	Specifies an IPv6 access list to match for redistributing routes that have been advertised by routers and access servers.

	Command or Action	Purpose
	`Router(config-route-map)# match ipv6 address acl1`	
Step 12	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 10`	Sets a tag value for the route in the route map.
Step 13	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map map1 permit 20`	Specifies route map name and set action to redistribute the route if the match criteria are met.
Step 14	**match interface** *interface-type interface-number* [*...interface-type interface-number*] **Example:** `Router(config-route-map)# match interface ethernet 0/0`	Specifies the next hop out of the interface to distribute the associated routes.
Step 15	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 20`	Sets a tag value for the route in the route map.
Step 16	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map map1 permit 30`	Specifies route map name and set action to redistribute the route if the match criteria are met.
Step 17	**match metric** *bandwidth delay reliability loading mtu* **Example:** `Router(config-route-map)# match metric 10000 100 255 100 1500`	Specifies the metric value for EIGRP in a route map.
Step 18	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map map2 permit 10`	Enters route-map configuration mode. • Specifies route map name and set action to redistribute the route if the match criteria are met.
Step 19	**match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*} **Example:** `Router(config-route-map)# match ipv6 address acl1`	Specifies an IPv6 access list to match for redistributing routes that have been advertised by routers and access servers.

	Command or Action	Purpose
Step 20	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 10`	Sets a tag value for the route in the route map.
Step 21	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map map2 permit 20`	Specifies route map name and set action to redistribute the route if the match criteria are met.
Step 22	**match interface** *interface-type interface-number* [*...interface-type interface-number*] **Example:** `Router(config-route-map)# match interface ethernet 0/0`	Specifies the next hop out of the interface to distribute the associated routes.
Step 23	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 20`	Sets a tag value for the route in the route map.
Step 24	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map map2 permit 30`	Specifies route map name and set action to redistribute the route if the match criteria are met.
Step 25	**match metric** *bandwidth delay reliability loading mtu* **Example:** `Router(config-route-map)# match metric 1000 100 255 200 1800`	Specifies the metric value for EIGRP in a route map.
Step 26	**end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns to privileged EXEC mode.

Configuration Examples for EIGRP Support for Route Map Filtering

Example Setting EIGRP Tags Using a Route Map--Autonomous System Configuration Examples

The following example shows how to configure a route map to match an EIGRP external protocol metric route with an allowable deviation of 100, a source protocol of BGP, and an autonomous system 45000. When the two match clauses are true, the tag value of the destination routing protocol is set to 5. The route map is used to distribute incoming packets for an EIGRP process.

```
Router(config)# route-map metric-range
Router(config-route-map)# match metric external 500 +- 100
Router(config-route-map)# match source-protocol bgp 45000
Router(config-route-map)# set tag 5
Router(config-route-map)# exit
Router(config)# router eigrp 1
Router(config-router)# network 172.16.0.0
Router(config-router)# distribute-list route-map metric_range in
```

The following example shows how to configure a route map to match EIGRP routes with a metric of 110, 200, or an inclusive range of 700 to 800. When the match clause is true, the tag value of the destination routing protocol is set to 10. The route map is used to redistribute EIGRP packets.

```
Router(config)# route-map metric-eigrp
Router(config-route-map)# match metric 110 200 750 +- 50
Router(config-route-map)# set tag 10
Router(config-route-map)# exit
Router(config)# router eigrp 1
Router(config-router)# network 172.21.1.0/24
Router(config-router)# redistribute eigrp route-map metric-eigrp
```

Example Setting EIGRP Tags Using a Route Map--Named Configuration Examples

The following example shows how to configure a route map to match an EIGRP external protocol metric route with an allowable deviation of 100, a source protocol of BGP, and an autonomous system 45000. When the two match clauses are true, the tag value of the destination routing protocol is set to 5. The route map is used to distribute incoming packets for an EIGRP process.

```
Router(config)# route-map metric_range
Router(config-route-map)# match metric external 500 +- 100
Router(config-route-map)# match source-protocol bgp 45000
Router(config-route-map)# set tag 5
Router(config-route-map)# exit
Router(config)# router eigrp virtual-name
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# network 172.21.1.0/24
```

```
Router(config-router-af)# topology base
Router(config-router-af-topology)# distribute-list route-map metric_range in
```

The following example shows how to configure a route map to match EIGRP routes with a metric of 110, 200, or an inclusive range of 700 to 800. When the match clause is true, the tag value of the destination routing protocol is set to 10. The route map is used to redistribute EIGRP packets.

```
Router(config)# route-map metric_eigrp
Router(config-route-map)# match metric 110 200 750 +- 50
Router(config-route-map)# set tag 10
Router(config-route-map)# exit
Router(config)# router eigrp virtual-name
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# network 172.21.1.0/24
Router(config-router-af)# topology base
Router(config-router-af-topology)# distribute-list route-map metric-range in
```

Example Configuring EIGRP Route-map for Distribute-list in IPv6

The following example shows how to configure EIGRP route maps for distribute list in IPv6.

```
enable
configure terminal
router eigrp test
 address-family ipv6 unicast autonomous-system 1
 topology base
 distribute-list route-map map_in
 distribute-list route-map map_out
 exit-af-topology
 exit-address-family
route-map map_in permit 10
 match ipv6 address acl1
 set tag 15
 route-map map_in permit 20
 match interface Ethernet0/0
 set tag 25
 route-map map_in permit 30
 match metric 10000 1000 255 255 1024
 route-map map_out permit 20
 match ipv6 address acl1
 set tag 25
 route-map map_out permit 40
 match interface Ethernet0/0
 set tag 35
 route-map map_out permit 50
 match metric 10000 100 255 200 1024
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
EIGRP overview and configuration	Configuring EIGRP
EIGRP commands including syntax, usage guidelines, and examples	*Cisco IOS IP Routing: EIGRP Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 171: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 140

EIGRP Route Tag Enhancements

The EIGRP Route Tag Enhancements feature enables you to specify and display route tags in dotted-decimal format, filter routes using the route tag value with wildcard mask, and set a default route tag for all internal Enhanced Interior Gateway Routing Protocol (EIGRP) routes.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for EIGRP Route Tag Enhancements

- Default route tags are not supported in EIGRP autonomous system configurations.
- Route tags will not be displayed in dotted-decimal format if the **route-tag notation** global configuration command is not enabled on the device.

Information About EIGRP Route Tag Enhancements

EIGRP Route Tag Enhancements Overview

A route tag is a 32-bit value attached to routes. Route tags are used to filter routes and apply administrative policies, such as redistribution and route summarization, to tagged routes. You can tag routes within a route map by using the **set tag** command. You can match tagged routes and apply administrative policies to tagged routes within a route map by using the **match tag** or **match tag list** command. The **match tag list** command is used to match a list of route tags.

Prior to the EIGRP Route Tag Enhancements feature, EIGRP routes could only be tagged using plain decimals (range: 1 to 4294967295). This feature enables users to specify and display route tag values as dotted decimals (range: 0.0.0.0 to 255.255.255.255), similar to the format used by IPv4 addresses. This enhancement is intended to simplify the use of route tags as users can now filter routes by using the route tag wildcard mask.

This feature also allows you to configure a default route tag for all internal EIGRP routes without using route maps. Use the **eigrp default-route-tag** command in address family configuration mode to configure a default route tag for internal EIGRP routes.

How to Configure EIGRP Route Tag Enhancements

Enabling Dotted-Decimal Notation for Route Tags

Perform this task to enable route tags to be displayed as dotted decimals in **show** commands, irrespective of whether or not the tags were configured as dotted decimals.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-tag notation dotted-decimal**
4. **end**
5. Enter one of the following:
   - **show ip route tag**
   - **show ipv6 route tag**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-tag notation dotted-decimal** **Example:** `Device(config)# route-tag notation dotted-decimal`	Enables the display of route tags in dotted-decimal format.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Step 5	Enter one of the following:
	• **show ip route tag**
	• **show ipv6 route tag**
	Example:
	`Device# show ip route tag`
	`Device# show ipv6 route tag`
		(Optional) Displays route tag entries for IPv4 or IPv6 routes.

Setting a Route Tag in a Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-name* **[permit | deny]** [*sequence-number*]
4. **set tag** {*tag-value* | *tag-value-dotted-decimal*}
5. **end**
6. **show route-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-name* **[permit | deny]** [*sequence-number*] **Example:** `Device(config)# route-map rip-to-eigrp`	Configures a route map and enters route-map configuration mode.
Step 4	**set tag** {*tag-value* | *tag-value-dotted-decimal*} **Example:** `Device(config-route-map)# set tag 7.7.7.7`	Sets a tag value for a route. **Note** In this example, all routes from Routing Information Protocol (RIP) to EIGRP are given a tag value of 7.7.7.7.
Step 5	**end** **Example:** `Device(config-route-map)# end`	Exits to privileged EXEC mode.
Step 6	**show route-map** **Example:** `Device# show route-map`	(Optional) Displays static and dynamic route maps configured on the router.

Matching a Route Tag in a Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-name* **[permit | deny]** [*sequence-number*]
4. **match tag** {*tag-value* | *tag-value-dotted-decimal*} [**. . .** *tag-value* | *tag-value-dotted-decimal*]
5. **end**
6. **show route-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-name* **[permit | deny]** [*sequence-number*] **Example:** `Device(config)# route-map eigrp-to-rip`	Configures a route map and enters route-map configuration mode.
Step 4	**match tag** {*tag-value* | *tag-value-dotted-decimal*} [**. . .** *tag-value* | *tag-value-dotted-decimal*] **Example:** `Device(config-route-map)# match tag 10.10.10.0`	Filters routes that match specific route tags.
Step 5	**end** **Example:** `Device(config-route-map)# end`	Exits to privileged EXEC mode.
Step 6	**show route-map** **Example:** `Device# show route-map`	(Optional) Displays static and dynamic route maps configured on the device.

Creating a Route Tag List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-dotted-decimal mask*
4. **end**
5. **show route-tag list** [*list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-dotted-decimal mask* **Example:** `Device(config)# route-tag list to-rip permit 10.10.10.0 0.0.0.7`	Creates a route tag list. • Route tag lists are used by route maps to match routes based on conditions specified in the route tag lists.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.
Step 5	**show route-tag list** [*list-name*] **Example:** `Device(config-router)# show route-tag list to-rip`	(Optional) Displays information about route tag lists configured on the device. • Use the *list-name* argument to display information about a specific route tag list.

Matching a Route Tag List

Route tag lists are used in route maps to match routes based on conditions specified in the route tag lists. Multiple route tag and mask pair sequences can be configured to permit or deny any condition for a list of route tags.

Note You can match either a route tag or a route tag list within a single route map sequence.

Perform this task to match routes based on conditions specified in the route tag list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-value-dotted-decimal mask*
4. **route-map** *map-name* **[permit | deny]** [*sequence-number*]
5. **match tag list** *list-name* [. . . *list-name*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-value-dotted-decimal mask* **Example:** `Device(config)# route-tag list list1 permit 10.10.10.0 0.0.0.7`	Configures a route tag list.
Step 4	**route-map** *map-name* **[permit | deny]** [*sequence-number*] **Example:** `Device(config)# route-map to-ospf`	Configures a route map and enters route-map configuration mode.
Step 5	**match tag list** *list-name* [. . . *list-name*] **Example:** `Device(config-route-map)# match tag list list1`	Filters routes that match a specified route tag list.
Step 6	**end** **Example:** `Device(config-route-map)# end`	Exits to privileged EXEC mode.

Setting a Default Route Tag for EIGRP Internal Routes

Perform this task to set a default route tag for all internal EIGRP routes without using a route map. Default route tags are supported only in EIGRP named mode configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. Enter one of the following:
   - **address-family ipv4 unicast autonomous-system** *autonomous-system-number*

- **address-family ipv6 unicast autonomous-system** *autonomous-system-number*

5. **eigrp default-route-tag** {*route-tag-plain-decimal* | *route-tag-dotted-decimal*}
6. **end**
7. Enter one of the following:
   - **show eigrp address-family ipv4 topology**
   - **show eigrp address-family ipv6 topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4 unicast autonomous-system** *autonomous-system-number*
	• **address-family ipv6 unicast autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 unicast autonomous-system 1`
	`Device(config-router)# address-family ipv6 unicast autonomous-system 1`
		Enters IPv4 or IPv6 address family configuration mode and configures an EIGRP routing instance.

Step 5	**eigrp default-route-tag** {*route-tag-plain-decimal* | *route-tag-dotted-decimal*} **Example:** `Device(config-router-af)# eigrp default-route-tag 10`	Sets a default route tag for all internal EIGRP routes.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits to privileged EXEC mode.

Step 7	Enter one of the following:
	• **show eigrp address-family ipv4 topology**
	• **show eigrp address-family ipv6 topology**
	Example:
	`Device(config-router-af)# show eigrp address-family ipv4 topology`
	`Device(config-router-af)# show eigrp address-family ipv6 topology`
		(Optional) Displays entries of EIGRP address-family IPv4 or IPv6 topology tables.

Configuration Examples for EIGRP Route Tag Enhancements

Example: Enabling Dotted-Decimal Notation for Route Tags

The following example shows how to enable the display of route tags in dotted-decimal format by using the **route-tag notation** command. If you do not configure the **route-tag notation** command, route tags will be displayed as plain decimals in **show** commands even if the route tags were configured as dotted decimals. When you configure the **route-tag notation** command, route tags will be displayed as dotted decimals even if the route tags were configured as plain decimals.

```
Device# configure terminal
Device(config)# route-tag notation dotted-decimal
```

Example: Setting a Route Tag

The following example shows how to redistribute EIGRP routes into RIP and RIP routes into EIGRP by setting tags for routes within route maps:

```
Device(config)# route-map eigrp-to-rip
Device(config-route-map)# set tag 10.10.10.10
Device(config-route-map)# exit
Device(config)# route-map rip-to-eigrp
Device(config-route-map)# set tag 20.20.20.20
Device(config-route-map)# exit
Device(config)# router rip
Device(config-router)# redistribute eigrp 7 route-map eigrp-to-rip metric 5
Device(config-router)# exit
Device(config)# router eigrp name
```

```
Device(config-router)# address-family ipv4 autonomous-system 10
Device(config-router-af)# topology base
Device(config-router-af-topology)# redistribute rip route-map rip-to-eigrp 2 2 2 2 2
Device(config-router-af-topology)# end
```

Example: Matching a Route Tag

The following example shows how to redistribute EIGRP routes with a route tag value of 10.10.10.10 into a RIP domain:

```
Device(config)# route-map eigrp-to-rip
Device(config-route-map)# match tag 10.10.10.10
Device(config-route-map)# exit
Device(config)# router rip
Device(config-router)# redistribute eigrp 7 route-map eigrp-to-rip 5
Device(config-router)# end
```

Example: Configuring a Route Tag List

The following example shows how to configure a route tag list named TAG with various criteria for filtering routes. Route maps will use this list to match routes based on the criteria specified in the list. Route tag lists can accept route tags and wild card masks.

```
Device(config)# route-tag list TAG permit 1.1.1.1 0.0.0.1
Device(config)# route-tag list TAG seq 3 permit 2.2.2.2 0.0.0.3
Device(config)# route-tag list TAG seq 10 permit 3.3.3.3 0.0.0.7
Device(config)# route-tag list TAG seq 15 5.5.5.5 0.0.0.31
Device(config)# route-tag list TAG seq 20 deny 4.4.4.4 0.0.0.4
```

Example: Matching a Route Tag List

The following example shows how to use a route map to filter routes that match a specific route tag list. A single list can have multiple match criteria. All criteria must match before the route can be filtered. This example shows how to configure a route tag list named List1 in a route map and use the **match tag list** command to filter routes that match the criteria listed in the route tag list.

```
Device(config)# route-tag list List1 permit 10.10.10.0 0.0.0.7
Device(config)# route-map to-ospf
Device(config-route-map)# match tag list List1
Device(config-route-map)# exit
Device(config)# router ospf 10
Device(config-router)# redistribute eigrp 7 route-map to-ospf metric 20
Device(config-router)# end
```

Example: Setting a Default Route Tag

The following example shows how to set a default route tag for all internal EIGRP routes without using a route map. Default route tags are supported only in EIGRP named configurations.

```
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 unicast autonomous-system 1
Device(config-router-af)# eigrp default-route-tag 10.10.10.10
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
EIGRP commands	EIGRP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 172: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 141

BFD Support for EIGRP IPv6

The BFD Support for EIGRP IPv6 feature provides Bidirectional Forwarding Detection (BFD) support for Enhanced Interior Gateway Routing Protocol (EIGRP) IPv6 sessions, thereby facilitating rapid fault detection and alternate-path selection in EIGRP IPv6 topologies. BFD is a detection protocol that provides a consistent failure-detection method for network administrators, and network administrators use BFD to detect forwarding path failures at a uniform rate and not at variable rates for different routing protocol 'Hello' mechanisms. This failure-detection methodology ensures easy network profiling and planning and consistent and predictable reconvergence time. This document provides information about BFD support for EIGRP IPv6 networks and explains how to configure BFD support in EIGRP IPv6 networks.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for BFD Support for EIGRP IPv6

EIGRP IPv6 sessions have a shutdown option in router, address family, and address-family interface configuration modes. To enable BFD support on EIGRP IPv6 sessions, the routing process should be in no shut mode in the abovementioned modes.

Restrictions for BFD Support for EIGRP IPv6

- The BFD Support for EIGRP IPv6 feature is supported only in EIGRP named mode.
- EIGRP supports only single-hop Bidirectional Forwarding Detection (BFD).
- The BFD Support for EIGRP IPv6 feature is not supported on passive interfaces.

Information About BFD Support for EIGRP IPv6

BFD for EIGRP IPv6

Bidirectional Forwarding Detection (BFD) is a detection protocol that provides fast-forwarding, path-failure detection for all media types, encapsulations, topologies, and routing protocols. The BFD Support for EIGRP IPv6 feature enables BFD to interact with the Enhanced Interior Gateway Routing Protocol (EIGRP) to create BFDv6 sessions between EIGRP neighbors. In a BFD-enabled EIGRP IPv6 session, BFD constantly monitors the forwarding path (from a local device to a neighboring device) and provides consistent failure detection at a uniform rate. Because failure detection happens at a uniform rate and not at variable rates, network profiling and planning is easier, and the reconvergence time remains consistent and predictable.

BFD is implemented in EIGRP at multiple levels; it can be implemented per interface or on all interfaces. When BFD is enabled on a specific interface, all peer relationships formed through the EIGRP “Hello” mechanism on that interface are registered with the BFD process. Subsequently, BFD establishes a session with each of the peers in the EIGRP topology and notifies EIGRP through a callback mechanism of any change in the state of any peer. When a peer is lost, BFD sends a “peer down” notification to EIGRP, and EIGRP unregisters a peer from BFD. BFD does not send a “peer up” notification to EIGRP when the peer is up because BFD now has no knowledge of the state of the peer. This behavior prevents rapid neighbor bouncing and repetitive route computations. The EIGRP “Hello” mechanism will later allow peer rediscovery and reregistration with the BFD process.

How to Configure BFD Support for EIGRP IPv6

Configuring BFD Support on All Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 address** *ipv6-address/prefix-length*
6. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
7. **exit**
8. **router eigrp** *virtual-name*

9. **address-family ipv6 autonomous-system** *as-number*
10. **eigrp router-id** *ip-address*
11. **af-interface default**
12. **bfd**
13. **end**
14. **show eigrp address-family ipv6 neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet0/0/1`	Specifies the interface type and number, and enters the interface configuration mode.
Step 5	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	Configures an IPv6 address.
Step 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Sets the baseline BFD session parameters on an interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 8	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Specifies an EIGRP routing process and enters router configuration mode.
Step 9	**address-family ipv6 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 3`	Enters address family configuration mode for IPv6 and configures an EIGRP routing instance.
Step 10	**eigrp router-id** *ip-address* **Example:** `Device(config-router-af)# eigrp router-id 172.16.1.3`	Sets the device ID used by EIGRP for this address family when EIGRP peers communicate with their neighbors.
Step 11	**af-interface default** **Example:** `Device(config-router-af)# af-interface default`	Configures interface-specific commands on all interfaces that belong to an address family in EIGRP named mode configurations, and enters address-family interface configuration mode.
Step 12	**bfd** **Example:** `Device(config-router-af-interface)# bfd`	Enables BFD on all interfaces.
Step 13	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address-family interface configuration mode and returns to privileged EXEC mode.
Step 14	**show eigrp address-family ipv6 neighbors** **Example:** `Device# show eigrp address-family ipv6 neighbors`	(Optional) Displays neighbors for which BFD has been enabled.

Configuring BFD Support on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 address** *ipv6-address /prefix-length*
6. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
7. **exit**

8. **router eigrp** *virtual-name*
9. **address-family ipv6 autonomous-system** *as-number*
10. **eigrp router-id** *ip-address*
11. **af-interface** *interface-type interface-number*
12. **bfd**
13. **end**
14. **show eigrp address-family ipv6 neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet0/0/1`	Specifies the interface type and number, and enters the interface configuration mode.
Step 5	**ipv6 address** *ipv6-address /prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	Configures an IPv6 address.
Step 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Sets the baseline BFD session parameters on an interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 8	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Specifies an EIGRP routing process and enters router configuration mode.
Step 9	**address-family ipv6 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 3`	Enters address family configuration mode for IPv6 and configures an EIGRP routing instance.
Step 10	**eigrp router-id** *ip-address* **Example:** `Device(config-router-af)# eigrp router-id 172.16.1.3`	Sets the device ID used by EIGRP for this address family when EIGRP peers communicate with their neighbors.
Step 11	**af-interface** *interface-type interface-number* **Example:** `Device(config-router-af)# af-interface gigabitethernet0/0/1`	Configures interface-specific commands on an interface that belongs to an address family in an EIGRP named mode configuration, and enters address-family interface configuration mode.
Step 12	**bfd** **Example:** `Device(config-router-af-interface)# bfd`	Enables BFD on the specified interface.
Step 13	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address-family interface configuration mode and returns to privileged EXEC mode.
Step 14	**show eigrp address-family ipv6 neighbors** **Example:** `Device# show eigrp address-family ipv6 neighbors`	(Optional) Displays neighbors for which BFD has been enabled.

Configuration Examples for BFD Support for EIGRP IPv6

Example: Configuring BFD Support on All Interfaces

```
Device(config)# ipv6 unicast-routing
Device(config)# interface GigabitEthernet0/0/1
Device(config-if)# ipv6 address 2001:0DB8:1::12/64
Device(config-if)# bfd interval 50 min_rx 50 multiplier 3
Device(config-if)# exit
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 unicast autonomous-system 1
```

```
Device(config-router-af)# eigrp router-id 172.16.0.1
Device(config-router-af)# af-interface default
Device(config-router-af-interface)# bfd
Device(config-router-af-interface)# end
```

Example: Configuring BFD Support on an Interface

```
Device(config)# ipv6 unicast-routing
Device(config)# GigabitEthernet0/0/1
Device(config-if)# ipv6 address 2001:DB8:A:B::1/64
Device(config-if)# bfd interval 50 min_rx 50 multiplier 3
Device(config-if)# exit
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 autonomous-system 3
Device(config-router-af)# af-interface GigabitEthernet0/0/1
Device(config-router-af-interface)# bfd
Device(config-router-af-interface)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
BFD commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	IP Routing: Protocol-Independent Command Reference
EIGRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	IP Routing: EIGRP Command Reference
Configuring EIGRP	"Configuring EIGRP" chapter in *IP Routing: EIGRP Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 173: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 142

EIGRP Loop-Free Alternate Fast Reroute

The EIGRP Loop-Free Alternate Fast Reroute feature allows the Enhanced Interior Gateway Routing Protocol (EIGRP) to reduce the routing transition time to less than 50 ms by precomputing repair paths or backup routes and installing these paths or routes in the Routing Information Base (RIB). Fast Reroute (FRR) is the mechanism that enables traffic that traverses a failed link to be rerouted around the failure. In EIGRP networks, precomputed backup routes or repair paths are known as feasible successors or loop-free alternates (LFAs). This module describes how to configure the EIGRP Loop-Free Alternate Fast Reroute feature and enable load-sharing and tie-breaking configurations for the feasible successors or LFAs that are identified by EIGRP.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for EIGRP Loop-Free Alternate Fast Reroute

- Only paths that are reachable through point-to-point interfaces are protected.
- IPv6 is not supported.

Information About EIGRP Loop-Free Alternate Fast Reroute

Repair Paths Overview

When a link or a device fails, distributed routing algorithms compute new routes or repair paths. The time taken for this computation is called routing transition. Until the transition is complete and all devices are converged on a common view of the network, the connectivity between the source and destination pairs of devices is interrupted. Repair paths forward traffic during a routing transition.

When a link or a device fails, initially only the neighboring devices are aware of the failure. All other devices in the network are unaware of the nature and location of this failure until information about this failure is propagated through the routing protocol. The propagation of this information may take several hundred milliseconds. Meanwhile, packets affected by the network failure need to be steered to their destinations. A device adjacent to the failed link employs a set of repair paths for packets that would have used the failed link. These repair paths are used from the time the router detects the failure until the routing transition is complete. By the time the routing transition is complete, all devices in the network revise their forwarding data and the failed link is eliminated from the routing computation. Routing protocols precompute repair paths in anticipation of failures so that the repair paths can be activated the moment a failure is detected. In Enhanced Interior Gateway Routing Protocol (EIGRP) networks, precomputed repair paths or backup routes are known as feasible successors or loop-free alternates (LFAs).

LFA Computation

A loop-free alternate (LFA) is a precomputed next-hop route that delivers a packet to its destination without looping back. Traffic is redirected to an LFA after a network failure and the LFA makes the forwarding decision without any knowledge of the failure.

Interior Gateway Protocols (IGPs) compute LFAs in the following two ways:

- Per-link (link-based) computation: In link-based LFAs, all prefixes (networks) that are reachable through the primary (protected) link share the same backup information. This means that the whole set of prefixes sharing the primary link also share the repair or the Fast Reroute (FRR) ability. The per-link approach protects only the next-hop address. It need not necessarily protect the destination node. Therefore, the per-link approach is suboptimal and not the best approach for capacity planning because all traffic from the primary link is redirected to the next hop instead of being spread over multiple paths. Redirecting all traffic to the next hop may lead to congestion on the link to the next hop
- Per-prefix (prefix-based) computation: Prefix-based LFAs allow computing backup information per prefix (network) and protect the destination address. The per-prefix approach is preferred over the per-link approach because of its greater applicability and better bandwidth utilization. Per-prefix computations provide better load sharing and better protection coverage than per-link computations because per-prefix computations evaluate all possible LFAs and use tie-breakers to select the best LFA from among the available LFAs.

Note The repair or backup information computed for a primary path by using prefix-based LFAs may be different from that computed by using link-based LFAs.

EIGRP always computes prefix-based LFAs. EIGRP uses the Diffusing Update Algorithm (DUAL) to calculate the successor and feasible successors. EIGRP uses the successor as the primary path and feasible successors as repair paths or LFAs.

LFA Tie-Breaking Rules

When there are multiple candidate LFAs for a given primary path, EIGRP uses a tie-breaking rule to select one LFA per primary path per prefix. A tie-breaking rule considers LFAs that satisfy certain conditions or have certain attributes. EIGRP uses the following four attributes to implement tie-breaking rules:

- Interface-disjoint—Eliminates LFAs that share the outgoing interface with the protected path.
- Linecard-disjoint—Eliminates LFAs that share the line card with the protected path.
- Lowest-repair-path-metric—Eliminates LFAs whose metric to the protected prefix is high. Multiple LFAs with the same lowest path metric may remain in the routing table after this tie-breaker is applied.
- Shared Risk Link Group (SRLG)-disjoint—Eliminates LFAs that belong to any of the protected path SRLGs. SRLGs refer to situations where links in a network share a common fiber (or a common physical attribute). If one link fails, other links in the group may also fail. Therefore, links in a group share risks.

How to Configure EIGRP Loop-Free Alternate Fast Reroute

Configuring LFA FRRs per Prefix

Perform this task to configure loop-free alternate (LFA) Fast Reroutes (FRRs) per prefix in an Enhanced Interior Gateway Routing Protocol (EIGRP) network. You can enable LFAs for all available prefixes in the EIGRP topology or for prefixes specified by route maps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *autonomous-system-number*
5. **topology base**
6. **fast-reroute per-prefix {all | route-map** *route-map-name***}**
7. **end**
8. **show ip eigrp topology frr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 1`	Enters IPv4 VRF address family configuration mode and configures an EIGRP routing instance.
Step 5	**topology base** **Example:** `Device(config-router-af)# topology base`	Configures a base EIGRP topology and enters router address family topology configuration mode.
Step 6	**fast-reroute per-prefix {all | route-map** *route-map-name***}** **Example:** `Device(config-router-af-topology)# fast-reroute per-prefix all`	Enables FRR for all prefixes in the topology. • Enter the **route-map** keyword to enable FRR on prefixes specified by a route map.
Step 7	**end** **Example:** `Device(config-router-af-topology)# end`	Exits router address family topology configuration mode and returns to privileged EXEC mode.
Step 8	**show ip eigrp topology frr** **Example:** `Device# show ip eigrp topology frr`	Displays the list of configured LFAs in the EIGRP topology table.

Disabling Load Sharing Among Prefixes

When the primary path is an Equal Cost Multipath (ECMP) path with multiple LFAs, prefixes (networks) are distributed equally among the LFAs because the default behavior for ECMP paths is load sharing. However, you can control the selection of LFAs by enabling tie-breaking configurations. To enable tie-breaking configurations, you should disable load sharing among prefixes. Perform this task to disable load sharing among prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *autonomous-system-number*

5. **topology base**
6. **fast-reroute load-sharing disable**
7. **end**
8. **show ip eigrp topology frr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 1`	Enters IPv4 VRF address family configuration mode and configures an EIGRP routing instance.
Step 5	**topology base** **Example:** `Device(config-router-af)# topology base`	Configures a base EIGRP topology and enters router address family topology configuration mode.
Step 6	**fast-reroute load-sharing disable** **Example:** `Device(config-router-af-topology)# fast-reroute load-sharing disable`	Disables load sharing among prefixes.
Step 7	**end** **Example:** `Device(config-router-af-topology)# end`	Exits router address family topology configuration mode and returns to privileged EXEC mode.
Step 8	**show ip eigrp topology frr** **Example:** `Device# show ip eigrp topology frr`	Displays the list of configured feasible successors or LFAs in the EIGRP topology table.

Enabling Tie-Breaking Rules for EIGRP LFAs

Perform this task to enable tie-breaking rules to select a single loop-free alternate (LFA) when there are multiple LFAs for a given primary path. The Enhanced Interior Gateway Routing Protocol (EIGRP) allows you to use four attributes to configure tie-breaking rules. Each of the following keywords of the **fast-reroute tie-break** command allows you to configure a tie-breaking rule based on a specific attribute: **interface-disjoint**, **linecard-disjoint**, **lowest-backup-path-metric**, and **srlg-disjoint**. You can assign a priority value for each attribute. Tie-breaking rules are applied on the basis of the priority assigned to each attribute. The lower the assigned priority value the higher the priority of the tie-breaking attribute.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *autonomous-system-number*
5. **topology base**
6. **fast-reroute tie-break {interface-disjoint | linecard-disjoint | lowest-backup-path-metric | srlg-disjoint}** *priority-number*
7. **end**
8. **show ip eigrp topology frr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 1`	Enters IPv4 VRF address family configuration mode and configures an EIGRP routing instance.
Step 5	**topology base** **Example:** `Device(config-router-af)# topology base`	Configures a base EIGRP topology and enters router address family topology configuration mode.

	Command or Action	Purpose
Step 6	**fast-reroute tie-break {interface-disjoint | linecard-disjoint | lowest-backup-path-metric | srlg-disjoint}** *priority-number* **Example:** `Device(config-router-af-topology)# fast-reroute tie-break lowest-backup-path-metric 2`	Enables EIGRP to select an LFA by configuring a tie-breaking attribute and assigning a priority to that attribute. • You cannot configure an attribute more than once in an address family.
Step 7	**end** **Example:** `Device(config-router-af-topology)# end`	Exits router address family topology configuration mode and returns to privileged EXEC mode.
Step 8	**show ip eigrp topology frr** **Example:** `Device# show ip eigrp topology frr`	Displays the list of configured feasible successors or LFAs in the EIGRP topology table.

Configuration Examples for EIGRP Loop-Free Alternate Fast Reroute

Example: Configuring LFA FRRs Per Prefix

The following example shows how to configure Enhanced Interior Gateway Routing Protocol (EIGRP) loop-free alternate (LFA) Fast Reroutes (FRRs) for prefixes specified by the route map named map1:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute per-prefix route-map map1
Device(config-router-af-topology)# end
```

Example: Disabling Load Sharing Among Prefixes

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute load-sharing disable
Device(config-router-af-topology)# end
```

Example: Enabling Tie-Breaking Rules

The following examples show how to enable tie-breaking configurations to allow the Enhanced Interior Gateway Routing Protocol (EIGRP) to select a loop-free alternate (LFA) when there are multiple candidate LFAs for a given primary path. The following example shows how to enable the tie-breaking rule that eliminates LFAs that share the outgoing interface with the primary path:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute tie-break interface-disjoint 2
Device(config-router-af-topology)# end
```

The following example shows how to enable the tie-breaking rule that eliminates LFAs that share the linecard with the primary path:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute tie-break linecard-disjoint 3
Device(config-router-af-topology)# end
```

The following example shows how to enable the tie-breaking rule that selects the LFA with the lowest metric to the the protected prefix:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute tie-break lowest-backup-path-metric 4
Device(config-router-af-topology)# end
```

The following example shows how to enable the tie-breaking rule that eliminates LFAs that share any SRLGs with the primary path:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 1
Device(config-router-af)# topology base
Device(config-router-af-topology)# fast-reroute tie-break srlg-disjoint 1
Device(config-router-af-topology)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases

Related Topic	Document Title
EIGRP commands	EIGRP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 174: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 143

Add Path Support in EIGRP

The Add Path Support in EIGRP feature enables hubs in a single Dynamic Multipoint VPN (DMVPN) domain to advertise multiple best paths to connected spokes when the Enhanced Interior Gateway Routing Protocol (EIGRP) is the routing protocol between the hubs and the spokes. This module provides information about the Add Path Support in EIGRP feature and explains how to configure it.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Add Path Support in EIGRP

All interfaces in an Enhanced Interior Gateway Routing Protocol (EIGRP) topology are by default configured with the **next-hop-self** command. This command enables EIGRP to set the local outbound interface as the next-hop value while advertising a route to a peer, even when advertising routes out of the interface on which the routes were learned. This default EIGRP behavior may interfere with the **add-paths**command that helps configure the Add Path Support in EIGRP feature. Therefore, before you configure this feature on a hub device in a Dynamic Multipoint VPN (DMVPN) domain, you must disable the **next-hop-self** command that is configured on the hub interface that connects to spokes in the DMVPN domain.

Restrictions for Add Path Support in EIGRP

- The Add Path Support in EIGRP feature can be enabled only in Enhanced Interior Gateway Routing Protocol (EIGRP) named mode configurations.
- The **variance** command should not be configured when the Add Path Support in EIGRP feature is enabled. The **variance** command alters the metrics of routes in an EIGRP topology, thereby enabling EIGRP to balance traffic among desired paths. Therefore, if you configure the **variance** command on a hub device, the command may interfere with the configuration of this feature.

Information About Add Path Support in EIGRP

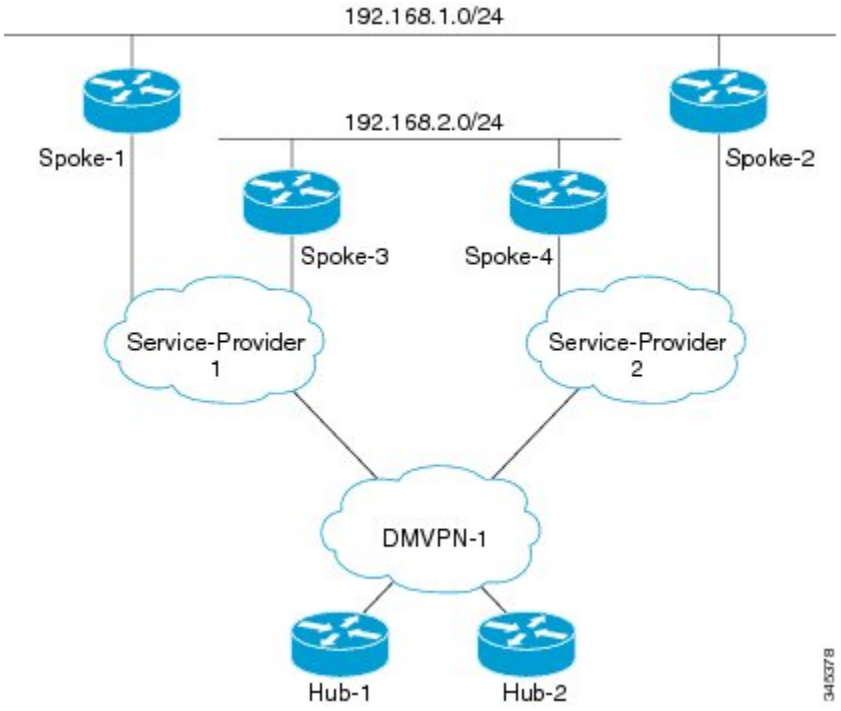
EIGRP Add Path Support Overview

In most Dynamic Multipoint VPN (DMVPN) domains, two or more spokes are connected to the same LAN segment. These spokes connect to more than one hub (for hub redundancy) through different service providers (for service-provider redundancy). In a single DMVPN domain, a hub connects to all spokes through one tunnel interface. In Enhanced Interior Gateway Routing Protocol (EIGRP) topologies, when a hub has more than one path (with the same metric but through different spokes) to reach the same network, both paths are chosen as best paths. However, by default, EIGRP advertises only one path as the best path to connected spokes. With the implementation of the Add Path Support in EIGRP feature, hubs in an EIGRP-DMVPN domain can advertise up to four additional best paths to connected spokes, thereby allowing load balancing and path redundancy. This feature supports both IPv4 and IPv6 configurations.

How Add Path Support in EIGRP Works

A typical single Dynamic Multipoint VPN (DMVPN) domain consists of dual hubs (for hub redundancy) connected to more than one service provider (for service-provider redundancy). In the figure below, two hub devices—Hub-1 and Hub-2—are connected through tunnel interfaces to a DMVPN domain.

Figure 134: Single DMVPN Domain



The DMVPN domain is in turn connected to two service providers—Service-Provider 1 and Service-Provider 2. Four spoke devices in this DMVPN domain—Spoke-1, Spoke-2, Spoke-3, and Spoke-4. Spoke-1 and Spoke-3 are connected to Service-Provider 1, and Spoke-2 and Spoke-4 are connected to Service-Provider 2. The Enhanced Interior Gateway Routing Protocol (EIGRP) is the routing protocol between the hubs and the spokes over the tunnel interfaces.

Spoke-1 and Spoke-2 are connected to a LAN with the network address 192.168.1.0/24. Both these spokes are connected to both the hubs through two different service providers, and hence, these spokes advertise the same LAN network to both hubs. Typically, spokes on the same LAN advertise the same metric; therefore, based on the metric, Hub-1 and Hub-2 have dual Equal-Cost Multipath (ECMP) routes to reach network 192.168.1.0/24. However, because EIGRP is a distance vector protocol, it advertises only one best path to the destination. Therefore, in this EIGRP-DMVPN domain, the hubs advertise only one route (for example, through Spoke-1) to reach network 192.168.1.0/24. When clients in subnet 192.168.2.0/24 communicate with clients in subnet 192.168.1.0/24, all traffic is directed to Spoke-1. Because of this default EIGRP behavior, there is no load balancing on Spoke-3 and Spoke-4. Additionally, if Spoke-1 fails or if the network of Service-Provider 1 goes down, EIGRP must reconverge to provide connectivity to 192.168.1.0/24.

The Add Path Support in EIGRP feature enables EIGRP to advertise up to four additional paths to connected spokes in a single DMVPN domain. If you configure this feature in the example topology discussed above, both Spoke-1 and Spoke-2 will be advertised to Spoke-3 and Spoke-4 as best paths to network 192.168.1.0, thereby allowing load balancing among all spokes in this DMVPN domain.

How to Configure Add Path Support in EIGRP

Configuring IPv4 Add Path Support on a Hub

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *as-number*
5. **af-interface** {**default** | *interface-type interface-number*}
6. **no next-hop-self** [**no-ecmp-mode**]
7. **add-paths** *number*
8. **end**
9. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 3`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 5	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface tunnel 0`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.

	Command or Action	Purpose
Step 6	**no next-hop-self** [**no-ecmp-mode**] **Example:** `Device(config-router-af-interface)# no next-hop-self no-ecmp-mode`	Instructs EIGRP to use the received next hop and not the local outbound interface address as the next hop to be advertised to neighboring devices.
Step 7	**add-paths** *number* **Example:** `Device(config-router-af-interface)# add-paths 4`	Enables EIGRP to advertise multiple paths as best paths to connected spokes in a single Dynamic Multipoint VPN (DMVPN) domain.
Step 8	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.
Step 9	**show running-config** **Example:** `Device# show running-config | section eigrp`	Displays contents of the current running configuration file. • Use the output modifier "|" to display the EIGRP section of the running configuration, and to verify whether the **add-paths** command is enabled in the configuration.

Configuring IPv6 Add Path Support on a Hub

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **router eigrp** *virtual-name*
5. **address-family ipv6 autonomous-system** *as-number*
6. **af-interface** {**default** | *interface-type interface-number*}
7. **no next-hop-self** [**no-ecmp-mode**]
8. **add-paths** *number*
9. **end**
10. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.
Step 5	**address-family ipv6 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 3`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 6	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface tunnel 0`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 7	**no next-hop-self** [**no-ecmp-mode**] **Example:** `Device(config-router-af-interface)# no next-hop-self no-ecmp-mode`	Instructs EIGRP to use the received next-hop address and not the local outbound interface address as the next hop to be advertised to neighboring devices.
Step 8	**add-paths** *number* **Example:** `Device(config-router-af-interface)# add-paths 4`	Enables EIGRP to advertise multiple paths as best paths to connected spokes in a single Dynamic Multipoint VPN (DMVPN) domain.
Step 9	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.
Step 10	**show running-config** **Example:** `Device# show running-config | section eigrp`	Displays contents of the current running configuration file. • Use the output modifier "|" to display the EIGRP section of the running configuration, and to verify whether the **add-paths** command is enabled in the configuration.

Configuration Examples for Add Path Support in EIGRP

Example: Configuring IPv4 Add Path Support on a Hub

```
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 10
Device(config-router-af)# af-interface tunnel 0
Device(config-router-af-interface)# no next-hop-self no-ecmp-mode
Device(config-router-af-interface)# add-paths 4
Device(config-router-af-interface)# end
```

Example: Configuring IPv6 Add Path Support on a Hub

```
Device(config)# ipv6 unicast-routing
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 autonomous-system 10
Device(config-router-af)# af-interface tunnel 0
Device(config-router-af-interface)# no next-hop-self no-ecmp-mode
Device(config-router-af-interface)# add-paths 4
Device(config-router-af-interface)# end
```

Additional References for Add Path Support in EIGRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP technology white papers	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 175: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 144

EIGRP Wide Metrics

The EIGRP Wide Metrics feature supports 64-bit metric calculations and Routing Information Base (RIB) scaling in Enhanced Interior Gateway Routing Protocol (EIGRP) topologies. The 64-bit calculations work only in EIGRP named mode configurations. EIGRP classic mode configurations use 32-bit calculations. This module provides an overview of the EIGRP Wide Metrics feature.

Information About EIGRP Wide Metrics

EIGRP Composite Cost Metrics

The Enhanced Interior Gateway Routing Protocol (EIGRP) uses bandwidth, delay, reliability, load, and K values (various constants that can be configured by a user to produce varying routing behaviors) to calculate the composite cost metric for local Routing Information Base (RIB) installation and route selections. The EIGRP composite cost metric is calculated using the following formula:

EIGRP composite cost metric = 256*((K1*Scaled Bw) + (K2*Scaled Bw)/(256 – Load) + (K3*Scaled Delay)*(K5/(Reliability + K4)))

EIGRP uses one or more vector metrics to calculate the composite cost metric. The table below lists EIGRP vector metrics and their descriptions.

Table 176: EIGRP Vector Metrics

Vector Metric	Description
bandwidth	The minimum bandwidth (Bw) of the route, in kilobits per second. It can be 0 or any positive integer. The bandwidth for the formula is scaled and inverted by using the following formula: Scaled Bw = (10^7/minimum bandwidth (Bw) in kilobits per second)
delay	Route delay, in tens of microseconds. Scaled Delay = (Delay/10)
load	The effective load of the route, expressed as a number from 0 to 255 (255 is 100 percent loading).

Vector Metric	Description
mtu	The minimum maximum transmission unit (MTU) size of the route, in bytes. It can be 0 or any positive integer.
reliability	The likelihood of successful packet transmission, expressed as a number between 0 and 255, where 255 means 100 percent reliability and 0 means no reliability.

EIGRP monitors metric weights, by using K values, on an interface to allow the tuning of EIGRP metric calculations and to indicate the type of service (ToS). K values are integers from 0 to 128; these integers, in conjunction with variables like bandwidth and delay, are used to calculate the overall EIGRP composite cost metric. The table below lists the K values and their defaults.

Table 177: EIGRP K-Value Defaults

Setting	Default Value
K1	1
K2	0
K3	1
K4	0
K5	0

Although you can configure K values to produce varying routing behaviors, most configurations use only the delay and bandwidth metrics by default, with bandwidth taking precedence, to produce a single 32-bit metric. Use of the default constants effectively reduces the above-mentioned composite cost metric formula to the following default formula: 256*(Scaled Bw + Scaled Delay).

For example, let us consider a link whose bandwidth to a particular destination is 128 kb/s and the delay is 84,000 microseconds. By using the default formula, you can simplify the EIGRP composite cost metric calculation to 256*(Scaled Bw + Scaled Delay), thus resulting in the following value:

Metric = $256*(10^7/128 + 84000/10) = 256*86525 = 22150400$

EIGRP Wide Metrics

The Enhanced Interior Gateway Routing Protocol (EIGRP) composite cost metric (calculated using the bandwidth, delay, reliability, load, and K values) is not scaled correctly for high-bandwidth interfaces or Ethernet channels, resulting in incorrect or inconsistent routing behavior. The lowest delay that can be configured for an interface is 10 microseconds. As a result, high-speed interfaces, such as 10 Gigabit Ethernet (GE) interfaces, or high-speed interfaces channeled together (GE ether channel) will appear to EIGRP as a single GE interface. This may cause undesirable equal-cost load balancing. To resolve this issue, the EIGRP Wide Metrics feature supports 64-bit metric calculations and Routing Information Base (RIB) scaling that provide the ability to support interfaces (either directly or via channeling techniques like port channels or ether channels) up to approximately 4.2 terabits.

Note The 64-bit metric calculations work only in EIGRP named mode configurations. EIGRP classic mode uses 32-bit metric calculations.

To accommodate interfaces with bandwidths above 1 gigabit and up to 4.2 terabits and to allow EIGRP to perform path selections, the EIGRP composite cost metric formula is modified. The paths are selected based on the computed time. The time that information takes to travel through links is measured in picoseconds. The interfaces can be directly capable of these high speeds, or the interfaces can be bundles of links with an aggregate bandwidth greater than 1 gigabit.

Metric = [(K1*Minimum Throughput + {K2*Minimum Throughput} / 256-Load) + (K3*Total Latency) + (K6*Extended Attributes)]* [K5/(K4 + Reliability)]

Default K values are as follows:

- K1 = K3 = 1
- K2 = K4 = K5 = 0
- K6 = 0

The EIGRP Wide Metrics feature also introduces K6 as an additional K value for future use.

By default, the path selection scheme used by EIGRP is a combination of throughput (rate of data transfer) and latency (time taken for data transfer), and the formula for calculating the composite cost metric is as follows:

Composite Cost Metric = (K1*Minimum Throughput) + (K3*Total Latency)

Minimum Throughput = (10^7* 65536)/Bw), where 65536 is the wide-scale constant.

Total Latency for bandwidths below 1 gigabit = (Delay*65536)/10, where 65536 is the wide-scale constant.

Total Latency for bandwidths above 1 gigabit = (10^7* 65536/10)/ Bw, 65536 is the wide-scale constant.

With the calculation of larger bandwidths, EIGRP can no longer fit the computed metric into a 4-byte unsigned long value that is needed by the Cisco RIB. To set the RIB scaling factor for EIGRP, use the **metric rib-scale** command. When you configure the **metric rib-scale** command, all EIGRP routes in the RIB are cleared and replaced with the new metric values.

EIGRP Metric Weights

You can use the **metric weights** command to adjust the default behavior of Enhanced Interior Gateway Routing Protocol (EIGRP) routing and metric computations. EIGRP metric defaults (K values) have been carefully selected to provide optimal performance in most networks.

Note Adjusting EIGRP metric weights can dramatically affect network performance. Because of the complexity of this task, we recommend that you do not change the default K values without guidance from an experienced network designer.

By default, the EIGRP composite cost metric is a 32-bit quantity that is the sum of segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. The formula used to scale and invert the bandwidth value is 10^7/minimum bandwidth in kilobits per second. However, with the EIGRP Wide Metrics

feature, the EIGRP composite cost metric is scaled to include 64-bit metric calculations for EIGRP named mode configurations.

For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Gigabit Ethernet (GE), and serial lines running from 9600 bits per second to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

Mismatched K Values

EIGRP K values are the metrics that EIGRP uses to calculate routes. Mismatched K values can prevent neighbor relationships from being established and can negatively impact network convergence. The example given below explains this behavior between two EIGRP peers (Device-A and Device-B).

The following configuration is applied to Device-A. The K values are changed using the **metric weights** command. A value of 2 is entered for the *k1* argument to adjust the bandwidth calculation. A value of 1 is entered for the *k3* argument to adjust the delay calculation.

```
Device(config)# hostname Device-A
Device-A(config)# interface serial 0
Device-A(config-if)# ip address 10.1.1.1 255.255.255.0
Device-A(config-if)# exit
Device-A(config)# router eigrp name1
Device-A(config-router)# address-family ipv4 autonomous-system 4533
Device-A(config-router-af)# network 10.1.1.0 0.0.0.255
Device-A(config-router-af)# metric weights 0 2 0 1 0 0 1
```

The following configuration is applied to Device-B, and the default K values are used. The default K values are 1, 0, 1, 0, 0, and 0.

```
Device(config)# hostname Device-B
Device-B(config)# interface serial 0
Device-B(config-if)# ip address 10.1.1.2 255.255.255.0
Device-B(config-if)# exit
Device-B(config)# router eigrp name1
Device-B(config-router)# address-family ipv4 autonomous-system 4533
Device-B(config-router-af)# network 10.1.1.0 0.0.0.255
Device-B(config-router-af)# metric weights 0 1 0 1 0 0 0
```

The bandwidth calculation is set to 2 on Device-A and set to 1 (by default) on Device-B. This configuration prevents these peers from forming a neighbor relationship.

The following error message is displayed on the console of Device-B because the K values are mismatched:

```
*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1 (Ethernet0/0) is
 down: K-value mismatch
```

The following are two scenarios where the above error message can be displayed:

- Two devices are connected on the same link and configured to establish a neighbor relationship. However, each device is configured with different K values.
- One of two peers has transmitted a “peer-termination” message (a message that is broadcast when an EIGRP routing process is shut down), and the receiving device does not support this message. The receiving device will interpret this message as a K-value mismatch.

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 178: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 145

EIGRP/SAF HMAC-SHA-256 Authentication

The EIGRP/SAF HMAC-SHA-256 Authentication feature enables packets in an Enhanced Interior Gateway Routing Protocol (EIGRP) topology or a Service Advertisement Framework (SAF) domain to be authenticated using Hashed Message Authentication Code-Secure Hash Algorithm-256 (HMAC-SHA-256) message authentication codes. This module discusses this feature from an EIGRP perspective; it gives a brief overview of this feature and explains how to configure it.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About EIGRP/SAF HMAC-SHA-256 Authentication

EIGRP Neighbor Relationship Maintenance

Neighbor relationship maintenance is the process that devices use to dynamically learn of other devices on their directly attached networks. Devices must also discover when their neighbors become unreachable or inoperative. Neighbor relationship maintenance is achieved with low overhead by devices when they periodically send small hello packets to each other. As long as hello packets are received, the Cisco software can determine whether a neighbor is alive and functioning. After the status of the neighbor is determined, neighboring devices can exchange routing information.

The reliable transport protocol is responsible for the guaranteed, ordered delivery of Enhanced Interior Gateway Routing Protocol (EIGRP) packets to all neighbors. The reliable transport protocol supports intermixed transmission of multicast and unicast packets. Some EIGRP packets (such as updates) must be sent reliably; this means that the packets require acknowledgment from the destination. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities, hello packets need not be sent reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello packet with an indication in the packet informing receivers that the packet need not be acknowledged. The reliable transport protocol can send multicast packets quickly when unacknowledged packets are pending, thereby ensuring that the convergence time remains low in the presence of varying speed links.

Some EIGRP remote unicast-listen (any neighbor that uses unicast to communicate) and remote multicast-group neighbors may peer with any device that sends a valid hello packet, thus causing security concerns. By authenticating the packets that are exchanged between neighbors, you can ensure that a device accepts packets only from devices that know the preshared authentication key.

HMAC-SHA-256 Authentication

Packets exchanged between neighbors must be authenticated to ensure that a device accepts packets only from devices that have the same preshared authentication key. Enhanced Interior Gateway Routing Protocol (EIGRP) authentication is configurable on a per-interface basis; this means that packets exchanged between neighbors connected through an interface are authenticated. EIGRP supports message digest algorithm 5 (MD5) authentication to prevent the introduction of unauthorized information from unapproved sources. MD5 authentication is defined in RFC 1321. EIGRP also supports the Hashed Message Authentication Code-Secure Hash Algorithm-256 (HMAC-SHA-256) authentication method. When you use the HMAC-SHA-256 authentication method, a shared secret key is configured on all devices attached to a common network. For each packet, the key is used to generate and verify a message digest that gets added to the packet. The message digest is a one-way function of the packet and the secret key. For more information on HMAC-SHA-256 authentication, see FIPS PUB 180-2, SECURE HASH STANDARD (SHS), for the SHA-256 algorithm and RFC 2104 for the HMAC algorithm.

If HMAC-SHA-256 authentication is configured in an EIGRP network, EIGRP packets will be authenticated using HMAC-SHA-256 message authentication codes. The HMAC algorithm takes as input the data to be authenticated (that is, the EIGRP packet) and a shared secret key that is known to both the sender and the receiver; the algorithm gives a 256-bit hash output that is used for authentication. If the hash value provided by the sender matches the hash value calculated by the receiver, the packet is accepted by the receiver; otherwise, the packet is discarded.

Typically, the shared secret key is configured to be identical between the sender and the receiver. To protect against packet replay attacks because of a spoofed source address, the shared secret key for a packet is defined as the concatenation of the user-configured shared secret (identical across all devices participating in the authenticated domain) with the IPv4 or IPv6 address (which is unique for each device) from which the packet is sent.

The device sending a packet calculates the hash to be sent based on the following:

- Key part 1—the configured shared secret.
- Key part 2—the local interface address from which the packet will be sent.
- Data—the EIGRP packet to be sent (prior to the addition of the IP header).

The device receiving the packet calculates the hash for verification based on the following:

- Key part 1—the configured shared secret.

- Key part 2—the IPv4 or IPv6 source address in the IPv4 or IPv6 packet header.
- Data—the EIGRP packet received (after removing the IP header).

For successful authentication, all of the following must be true:

- The sender and receiver must have the same shared secret.
- The source address chosen by the sender must match the source address in the IP header that the receiver receives.
- The EIGRP packet data that the sender transmits must match the EIGRP packet data that the receiver receives.

Authentication cannot succeed if any of the following is true:

- The sender does not know the shared secret expected by the receiver.
- The IP source address in the IP header is modified in transit.
- Any of the EIGRP packet data is modified in transit.

How to Configure EIGRP/SAF HMAC-SHA-256 Authentication

Configuring HMAC-SHA-256 Authentication

Before you begin

Perform this task to configure an interface to use basic Hashed Message Authentication Code-Secure Hash Algorithm-256 (HMAC-SHA-256) authentication with an encrypted password—password1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **af-interface** {**default** | *interface-type interface-number*}
7. **authentication mode** {**hmac-sha-256** *encryption-type password* | **md5**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 46000`
		Enters IPv4 or IPv6 VRF address family configuration mode and configures an EIGRP routing instance.

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router-af)# network 172.16.0.0`	Associates a network with an EIGRP routing process. **Note** This command is used only while configuring an IPv4 routing instance.
Step 6	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface ethernet 0/0`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 7	**authentication mode** {**hmac-sha-256** *encryption-type password* | **md5**} **Example:** `Device(config-router-af-interface)# authentication mode hmac-sha-256 7 password1`	Specifies the type of authentication to be used in an EIGRP address family for the EIGRP instance. In this case, the HMAC-SHA-256 authentication method is used.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to global configuration mode.

Configuration Examples for EIGRP/SAF HMAC-SHA-256 Authentication

Example: Configuring HMAC-SHA-256 Authentication

The following example shows how to configure Hashed Message Authentication Code-Secure Hash Algorithm-256 (HMAC-SHA-256) authentication with password password1.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name1
Device(config-router)# address-family ipv6 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# authentication mode hmac-sha-256 0 password1
Device(config-router-af-interface)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol

Standards and RFCs

Standard/RFC	Title
FIPS PUB 180-2	*SECURE HASH STANDARD (SHS)*

Standard/RFC	Title
RFC 1321	*The MD5 Message-Digest Algorithm*
RFC 2104	*HMAC: Keyed-Hashing for Message Authentication*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 179: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 146

IP EIGRP Route Authentication

The IP Enhanced IGRP Route Authentication feature provides MD5 authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About IP EIGRP Route Authentication

EIGRP Route Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

Each key has its own key identifier (specified with the **key** *number* key chain configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and the MD5 authentication key in use.

You can configure multiple keys with specific lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in the order from lowest to highest, and

uses the first valid key that it encounters. Note that the device needs to know the time to configure keys with lifetimes.

How to Configure IP EIGRP Route Authentication

Defining an Autonomous System for EIGRP Route Authentication

Before you begin

Before you configure EIGRP route authentication, you must enable EIGRP. In this task, EIGRP is defined with an autonomous system number.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no switchport**
5. **ip authentication mode eigrp** *autonomous-system* **md5**
6. **ip authentication key-chain eigrp** *autonomous-system key-chain*
7. **exit**
8. **key chain** *name-of-chain*
9. **key** *key-id*
10. **key-string** *text*
11. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
12. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0/9`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	Puts an interface into Layer 3 mode
Step 5	**ip authentication mode eigrp** *autonomous-system* **md5** **Example:** `Device(config-if)# ip authentication mode eigrp 1 md5`	Enables MD5 authentication in EIGRP packets.
Step 6	**ip authentication key-chain eigrp** *autonomous-system key-chain* **Example:** `Device(config-if)# ip authentication key-chain eigrp 1 keychain1`	Enables authentication of EIGRP packets.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	**key chain** *name-of-chain* **Example:** `Device(config)# key chain keychain1`	Identifies a key chain and enters key chain configuration mode.
Step 9	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Identifies the key number and enters key chain key configuration mode.
Step 10	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string 0987654321`	Identifies the key string.
Step 11	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite`	(Optional) Specifies the time period during which the key can be received.
Step 12	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:**	(Optional) Specifies the time period during which the key can be sent.

	Command or Action	Purpose
	`Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite`	
Step 13	**end** **Example:** `Device(config-keychain-key)# end`	Exits key chain key configuration mode and returns to privileged EXEC mode.

Defining a Named Configuration for EIGRP Route Authentication

Before you begin

Before you configure EIGRP route authentication, you must enable EIGRP. In this task, EIGRP is defined with a virtual instance name.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **af-interface** {**default** | *interface-type interface-number*}
7. **authentication key-chain** *name-of-chain*
8. **authentication mode** {**hmac-sha-256** *encryption-type password* | **md5**}
9. **exit-af-interface**
10. **exit-address-family**
11. **exit**
12. **key chain** *name-of-chain*
13. **key** *key-id*
14. **key-string** *text*
15. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
16. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router-af)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 6	**af-interface** {**default** | *interface-type interface-number*} **Example:**	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 7	**authentication key-chain** *name-of-chain* **Example:** `Device(config-router-af-interface)# authentication key-chain SITE1`	Specifies an authentication key chain for EIGRP.
Step 8	**authentication mode** {**hmac-sha-256** *encryption-type password* | **md5**} **Example:** `Device(config-router-af-interface)# authentication mode md5`	Specifies the type of authentication used in an EIGRP address family for the EIGRP instance.

	Command or Action	Purpose
Step 9	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 12	**key chain** *name-of-chain* **Example:** `Device(config)# key chain keychain1`	Identifies a key chain and enters key chain configuration mode.
Step 13	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Identifies the key number and enters key chain key configuration mode.
Step 14	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string 0987654321`	Identifies the key string.
Step 15	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite`	(Optional) Specifies the time period during which the key can be received.
Step 16	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite`	(Optional) Specifies the time period during which the key can be sent.
Step 17	**end** **Example:**	Exits key chain key configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-keychain-key)# end`	

Configuration Examples for IP EIGRP Route Authentication

Example: EIGRP Route Authentication—Autonomous System Definition

The following example shows how to enable MD5 authentication on EIGRP packets in autonomous system 1.

Device A will accept and attempt to verify the MD5 digest of any EIGRP packet with a key equal to 1. It will also accept a packet with a key equal to 2. All other MD5 packets will be dropped. Device A will send all EIGRP packets with key 2.

Device B will accept key 1 or key 2 and will use key 1 to send MD5 authentication because key 1 is the first valid key of the key chain. Key 1 is not valid after December 4, 2006. After this date, key 2 is used to send MD5 authentication, and this key is valid until January 4, 2007.

The figure below shows the scenario.

Device A Configuration

```
Device> enable
Device(config)# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface Gigabitethernet 1/0/9
Device(config-if)# no switchport
Device(config-if)# ip authentication mode eigrp 1 md5
Device(config-if)# ip authentication key-chain eigrp 1 key1
Device(config-if)# exit
Device(config)# key chain key1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 04:48:00 Dec 4 1996
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string 1234567890
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:45:00 Jan 4 2007 infinite
```

Device B Configuration

```
Device> enable
Device(config)# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface Gigabitethernet 1/0/9
Device(config-if)# no switchport
Device(config-if)# ip authentication mode eigrp 1 md5
Device(config-if)# ip authentication key-chain eigrp 1 key2
```

```
Device(config-if)# exit
Device(config)# key chain key2
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string 1234567890
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:45:00 Jan 4 2007 infinite
```

Example: EIGRP Route Authentication—Named Configuration

The following example shows how to enable MD5 authentication on EIGRP packets in a named configuration.

Device A will accept and attempt to verify the MD5 digest of any EIGRP packet with a key equal to 1. It will also accept a packet with a key equal to 2. All other MD5 packets will be dropped. Device A will send all EIGRP packets with key 2.

Device B will accept key 1 or key 2 and will use key 1 to send MD5 authentication because key 1 is the first valid key of the key chain. Key 1 is not valid after December 4, 2006. After this date, key 2 will be used to send MD5 authentication because it is valid until January 4, 2007.

Device A Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
Device(config-router-af)# af-interface Gigabitethernet 1/0/1
Device(config-router-af-interface)# authentication key-chain SITE1
Device(config-router-af-interface)# authentication mode md5
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af)# exit-address-family
Device(config-router)# exit
Device(config)# key chain SITE1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string 1234567890
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:45:00 Jan 4 2007 infinite
```

Device B Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name2
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# authentication key-chain SITE2
Device(config-router-af-interface)# authentication mode md5
```

```
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af)# exit-address-family
Device(config-router)# exit
Device(config)# key chain SITE2
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
```

The following example shows how to configure advanced SHA authentication with password password1 and several key strings that will be rotated as time passes:

```
!
key chain chain1
 key 1
  key-string securetraffic
  accept-lifetime 04:00:00 Dec 4 2006 infinite
  send-lifetime 04:00:00 Dec 4 2010 04:48:00 Dec 4 2008
 !
 key 2
  key-string newertraffic
  accept-lifetime 01:00:00 Dec 4 2010 infinite
  send-lifetime 03:00:00 Dec 4 2010 infinite
 exit
!
router eigrp virtual-name
  address-family ipv6 autonomous-system 4453
    af-interface ethernet 0
       authentication mode hmac-sha-256 0 password1
       authentication key-chain key1
  !
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 180: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 147

EIGRP IPv6 VRF-Lite

The EIGRP IPv6 VRF-Lite feature provides EIGRP IPv6 support for multiple VRFs and simplifies the management and troubleshooting of traffic belonging to a specific VRF.

Note The EIGRP IPv6 VRF-Lite feature is available only in EIGRP named configurations.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About EIGRP IPv6 VRF-Lite

VRF-Lite for EIGRP IPv6

The EIGRP IPv6 VRF-Lite feature provides separation between routing and forwarding, which supports an additional level of security because communication between devices belonging to different VRFs is not allowed, unless explicitly configured. While the EIGRP IPv6 VRF-Lite feature supports multiple VRFs, the feature also simplifies the management and troubleshooting of traffic belonging to a specific VRF.

Virtual Private Networks (VPNs) provide a secure way for customers to share bandwidth over a service provider backbone network. A VPN is a collection of sites sharing a common routing table. A customer site

is connected to the service provider network by one or more interfaces, and the service provider associates each interface with a VPN routing table. A VPN routing table is called a VPN routing/forwarding (VRF) table.

VRF-lite allows a service provider to support two or more VPNs with an overlapping IP address using one interface. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but a Layer 3 interface cannot belong to more than one VRF at any time.

Note The EIGRP IPv6 VRF-Lite feature is available only in EIGRP named configurations.

EIGRP Named Configuration

Configuring the **router eigrp** command with the *virtual-instance-name* argument creates an EIGRP configuration referred to as the EIGRP named configuration or EIGRP named mode. An EIGRP named configuration does not create an EIGRP routing instance by itself; it is a base configuration that is required to define address-family configurations that are used for routing.

In EIGRP named configurations, EIGRP VPNs can be configured in IPv4 and IPv6 named configurations. A VRF instance and a route distinguisher must be defined before the address family session can be created.

A single EIGRP routing process can support multiple VRFs. The number of VRFs that can be configured is limited only by the available system resources on the device, which is determined by the number running processes and available memory. However, only a single VRF can be supported by each VPN, and redistribution between different VRFs is not supported.

How to Configure EIGRP IPv6 VRF-Lite

Enabling the EIGRP IPv6 VRF-Lite Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Configures the EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# address-family ipv6 vrf vrf1 autonomous-system 45000`	Enables EIGRP IPv6 VRF-Lite and enters address family configuration mode.
Step 5	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for EIGRP IPv6 VRF-Lite

Example: Enabling EIGRP IPv6 VRF-Lite—Named Configuration

The following example shows how to enable the EIGRP IPv6 VRF-lite feature:

```
Device> enable
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# rd 100:1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv6 vrf vrf1 autonomous-system 45000
Device(config-router-af)#
```

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 181: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 148

EIGRP Stub Routing

The EIGRP stub routing feature improves network stability, reduces resource utilization, and simplifies the stub device configuration.

Stub routing is commonly used in hub-and-spoke network topologies. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote device (the spoke) that is connected to one or more distribution devices (the hub). The remote device is adjacent to one or more distribution devices. The only route for IP traffic to reach the remote device is through a distribution device.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About EIGRP Stub Routing

EIGRP Stub Routing

The EIGRP stub routing feature improves network stability, reduces resource utilization, and simplifies the stub device configuration.

Stub routing is commonly used in hub-and-spoke network topologies. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote device (the spoke) that is connected to one or more distribution devices (the hub). The remote device is adjacent to one or more distribution devices. The only route for IP traffic to reach the remote device is through a distribution device. This type of configuration is

commonly used in WAN topologies, where the distribution device is directly connected to a WAN. The distribution device can be connected to many remote devices, which is often the case. In a hub-and-spoke topology, the remote device must forward all nonlocal traffic to a distribution device, so it becomes unnecessary for the remote device to have a complete routing table. Generally, the distribution device need not send anything more than a default route to the remote device.

When using the EIGRP stub routing feature, you need to configure the distribution and remote devices to use EIGRP and configure only the remote device as a stub. Only specified routes are propagated from the remote (stub) device. The stub device responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message "inaccessible." A device that is configured as a stub will send a special peer information packet to all neighboring devices to report its status as a stub device.

Any neighbor that receives a packet informing it of the stub status will not query the stub device for any routes, and a device that has a stub peer will not query that peer. The stub device will depend on the distribution device to send proper updates to all peers.

The figure below shows a simple hub-and-spoke network.

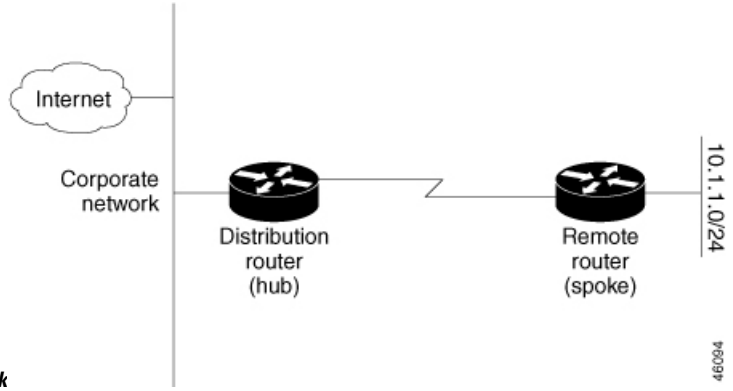


Figure 136: Simple Hub-and-Spoke Network

The stub routing feature by itself does not prevent routes from being advertised to the remote device. In the above example, the remote device can access the corporate network and the Internet only through the distribution device. Having a complete route table on the remote device would serve no functional purpose because the path to the corporate network and the Internet would always be through the distribution device. The large route table would only reduce the amount of memory required by the remote device. Bandwidth and memory can be conserved by summarizing and filtering routes in the distribution device. The remote device need not receive routes that have been learned from other networks because the remote device must send all nonlocal traffic, regardless of the destination, to the distribution device. If a true stub network is desired, the distribution device should be configured to send only a default route to the remote device. The EIGRP stub routing feature does not automatically enable summarization on distribution devices. In most cases, the network administrator will need to configure summarization on distribution devices.

Note When configuring the distribution device to send only a default route to the remote device, you must use the **ip classless** command on the remote device. By default, the **ip classless** command is enabled in all Cisco images that support the EIGRP stub routing feature.

Without the EIGRP stub routing feature, even after routes that are sent from the distribution device to the remote device have been filtered or summarized, a problem might occur. If a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution device, which in turn would send a query to the remote device, even if routes are being summarized. If there is a communication problem (over the WAN

link) between the distribution device and the remote device, an EIGRP stuck in active (SIA) condition could occur and cause instability elsewhere in the network. The EIGRP stub routing feature allows a network administrator to prevent queries from being sent to the remote device.

Dual-Homed Remote Topology

In addition to a simple hub-and-spoke network, where a remote device is connected to a single distribution device, the remote device can be dual-homed to two or more distribution devices. This configuration adds redundancy and introduces unique issues, and the stub feature helps to address some of these issues.

A dual-homed remote device will have two or more distribution (hub) devices. However, the principles of stub routing are the same as they are with a hub-and-spoke topology. The figure below shows a common dual-homed remote topology with one remote device: however, 100 or more devices could be connected on the same interfaces on distribution Device 1 and distribution Device 2. The remote device will use the best route to reach its destination. If distribution Device 1 experiences a failure, the remote device can still use distribution Device 2 to reach the corporate network.

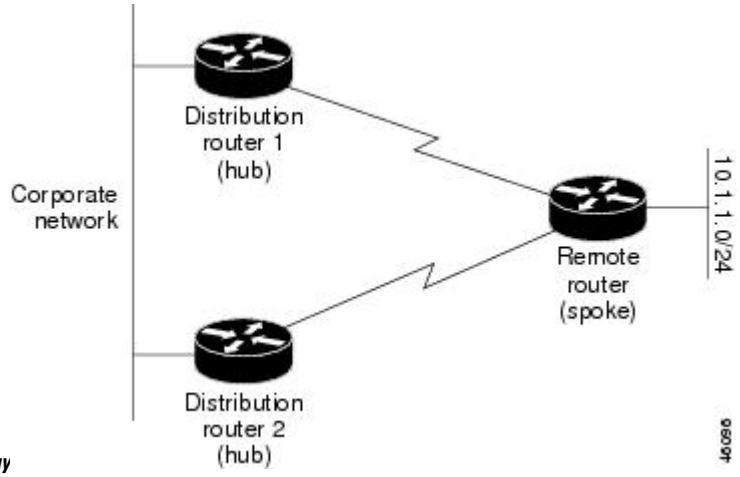
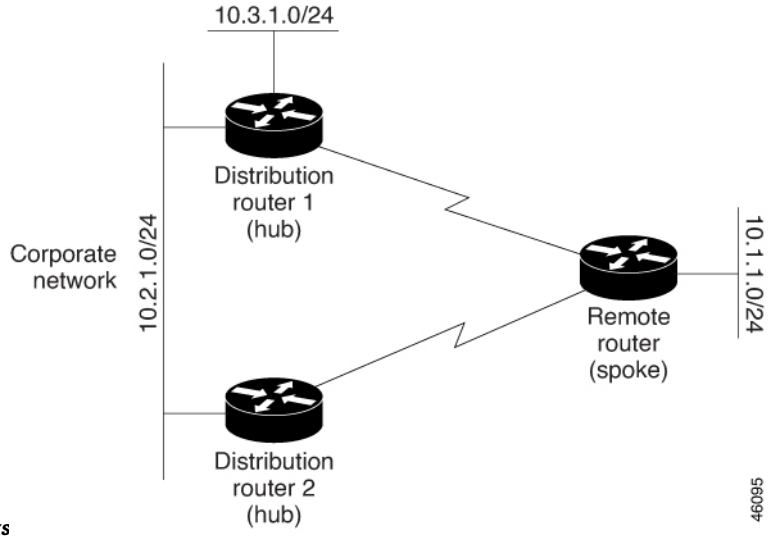


Figure 137: Simple Dual-Homed Remote Topology

The figure above shows a simple dual-homed remote topology with one remote device and two distribution devices. Both distribution devices maintain routes to the corporate network and stub network 10.1.1.0/24.

Dual-homed routing can introduce instability into an EIGRP network. In the figure below, distribution Device 1 is directly connected to network 10.3.1.0/24. If summarization or filtering is applied on distribution Device 1, the device will advertise network 10.3.1.0/24 to all of its directly connected EIGRP neighbors (distribution Device 2 and the remote device).

Figure 138: Dual-Homed Remote Topology with Distribution Device 1 Connected to Two

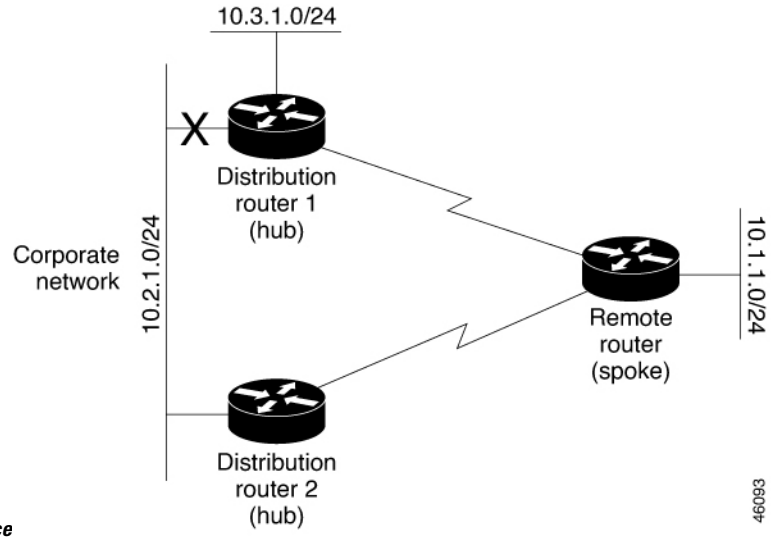


Networks

The figure above shows a simple dual-homed remote topology, where distribution Device 1 is connected to both network 10.3.1.0/24 and network 10.2.1.0/24.

If the 10.2.1.0/24 link between distribution Device 1 and distribution Device 2 fails, the lowest cost path to network 10.3.1.0/24 from distribution Device 2 will be through the remote device (see the figure below). This route is not desirable because the traffic that was previously traveling across the corporate network 10.2.1.0/24 would now be sent across a much lower bandwidth connection. The overutilization of the lower bandwidth WAN connection can cause many problems that might affect the entire corporate network. The use of the lower bandwidth route that passes through the remote device may cause WAN EIGRP distribution devices to be dropped. Serial lines on distribution and remote devices may also be dropped, and EIGRP SIA errors on the distribution and core devices can occur.

Figure 139: Dual-Homed Remote Topology with a Failed Route to a Distribution



Device

It is not desirable for traffic from distribution Device 2 to travel through any remote device to reach network 10.3.1.0/24. Backup routes can be used if links are sized to manage the load. However, most networks, of the type shown in the figure above, have remote devices located at remote offices with relatively slow links. To

ensure that traffic from distribution devices are not routed through a remote device, you can configure route summarization on the distribution device and the remote device.

It is typically undesirable for traffic from a distribution device to use a remote device as a transit path. A typical connection from a distribution device to a remote device would have much less bandwidth than a connection at the network core. Attempting to use a remote device with a limited bandwidth connection as a transit path would generally produce excessive congestion at the remote device. The EIGRP stub routing feature can prevent this problem by preventing the remote device from advertising core routes back to the distribution devices. In the above example, routes learned by the remote device from distribution Device 1 will not be advertised to distribution Device 2. Therefore, distribution Device 2 will not use the remote device as a transit for traffic destined to the network core.

The EIGRP stub routing feature provides network stability. If the network is not stable, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit devices. Instead, distribution devices to which the stub device is connected answer queries on behalf of the stub device. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP stub routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote devices to prevent those devices from appearing as transit paths to hub devices.

Caution The EIGRP stub routing feature should be used only on stub devices. A stub device is defined as a device connected to the network core or distribution layer through which core transit traffic should not flow. A stub device should not have any EIGRP neighbors other than distribution devices. Ignoring this restriction will cause undesirable behavior.

Note Multiaccess interfaces such as ATM, Gigabit Ethernet, Frame Relay, ISDN PRI, and X.25 are supported by the EIGRP stub routing feature only when all devices on that interface, except the hub, are configured as stub devices.

How to Configure EIGRP Stub Routing

Configuring the EIGRP Stub Routing Autonomous System Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **network** *ip-address* [**wildcard-mask**]
5. **eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**]
6. **end**
7. **show ip eigrp neighbors** [*interface-type* | *as-number* | **static** | **detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 1`	Configures a remote or distribution device to run an EIGRP process and enters router configuration mode.
Step 4	**network** *ip-address* [**wildcard-mask**] **Example:** `Device(config-router)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution device.
Step 5	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:** `Device(config-router)# eigrp stub connected static`	Configures a remote device as an EIGRP stub device.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 7	**show ip eigrp neighbors** [*interface-type* | *as-number* | **static** | **detail**] **Example:** `Device# show ip eigrp neighbors detail`	(Optional) Verifies that a remote device has been configured as a stub device with EIGRP. • Enter this command on the distribution device. The last line of the output displays the stub status of the remote or spoke device.

Configuring the EIGRP Stub Routing Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*

4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [**wildcard-mask**]
6. **eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**]
7. **exit-address-family**
8. **end**
9. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] [**neighbors**] [**static**] [**detail**] [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**network** *ip-address* [**wildcard-mask**] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution device.
Step 6	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:**	Configures a device as a stub using EIGRP.

	Command or Action	Purpose
	`Device(config-router-af) eigrp stub leak-map map1`	
Step 7	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 9	**show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] [**neighbors**] [**static**] [**detail**] [*interface-type interface-number*] **Example:** `Device# show eigrp address-family ipv4 neighbors detail`	(Optional) Displays neighbors discovered by EIGRP.

Configuration Examples for EIGRP Stub Routing

Example: EIGRP Stub Routing—Autonomous System Configuration

A device that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor devices by default. The following six keywords can be used with the **eigrp stub** command to modify this behavior:

- **connected**
- **leak-map**
- **receive-only**
- **redistributed**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command for an EIGRP autonomous system configuration.

Example: eigrp stub Command

In the following example, the **eigrp stub** command is used to configure the device as a stub that advertises connected and summary routes:

```
Device(config)# router eigrp 1
```

```
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub
```

Example: eigrp stub connected static Command

In the following example, the **eigrp stub** command is used with the **connected** and **static** keywords to configure the device as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub connected static
```

Example: eigrp stub leak-map Command

In the following example, the **eigrp stub** command is issued with the **leak-map** *name* keyword-argument pair to configure the device to reference a leak map that identifies routes that would have been suppressed:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub leak-map map1
```

Example: eigrp stub receive-only Command

In the following example, the **eigrp stub** command is issued with the **receive-only** keyword to configure the device as a receive-only neighbor (connected, summary, and static routes will not be sent):

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub receive-only
```

Example: eigrp stub redistributed Command

In the following example, the **eigrp stub** command is issued with the **redistributed** keyword to configure the device to advertise other protocols and autonomous systems:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub redistributed
```

Example: EIGRP Stub Routing—Named Configuration

A device that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor devices by default. The following six keywords can be used with the **eigrp stub** command to modify this behavior:

- **connected**

- **leak-map**
- **receive-only**
- **redistributed**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command for an EIGRP named configuration.

Example: eigrp stub Command

In the following example, the **eigrp stub** command is used to configure the device as a stub that advertises connected and summary routes:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af) eigrp stub
```

Example: eigrp stub connected static Command

In the following named configuration example, the **eigrp stub** command is issued with the **connected** and **static** keywords to configure the device as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub connected static
```

Example: eigrp stub leak-map Command

In the following named configuration example, the **eigrp stub** command is issued with the **leak-map** *name* keyword-argument pair to configure the device to reference a leak map that identifies routes that would normally have been suppressed:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub leak-map map1
```

Example: eigrp stub receive-only Command

In the following named configuration example, the **eigrp stub** command is issued with the **receive-only** keyword to configure the device as a receive-only neighbor (connected, summary, and static routes will not be sent):

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub receive-only
```

Example: eigrp stub redistributed Command

In the following named configuration example, the **eigrp stub** command is issued with the **redistributed** keyword to configure the device to advertise other protocols and autonomous systems:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af) eigrp stub redistributed
```

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 182: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 149

EIGRP Support for 6PE/6VPE

The EIGRP Support for 6PE/6VPE feature enables native IPv6 Enhanced Interior Gateway Routing Protocol (EIGRP) routes to preserve their original characteristics (metric and other attributes like type, delay, bandwidth, and maximum transmission unit [MTU]) while being redistributed from one IPv6 EIGRP site to another over a service-provider VPN cloud or an IPv6 provider edge (6PE) Multiprotocol Label Switching-VPN (MPLS-VPN) network. The Border Gateway Protocol (BGP) is used as the external routing protocol to transfer IPv6 EIGRP routes across the VPN cloud or the 6PE MPLS-VPN network. This module explains the EIGRP 6PE/6VPE feature.

Information About EIGRP Support for 6PE/6VPE

BGP Extended Communities

For the Enhanced Interior Gateway Routing Protocol (EIGRP) to recreate route metrics derived from the originating customer site, the original metrics are encoded into Border Gateway Protocol (BGP) Extended Communities by the provider-edge (PE) device that receives the routes from the transmitting customer-edge (CE) device. These extended communities are then transported across the Multiprotocol Label Switching-VPN (MPLS-VPN) backbone by BGP from one customer site to the other (peering customer site). After the peering customer site receives the routes, BGP redistributes the routes into EIGRP. EIGRP, then, extracts the BGP Extended Community information and reconstructs the routes as they appeared in the original customer site.

The following rules govern BGP Extended Communities:

Non-EIGRP-Originated Routes: If a non-EIGRP-originated route is received through BGP and the route has no extended community information for EIGRP, BGP advertises the route to the receiving CE as an external EIGRP route by using the route's default metric. If no default metric is configured, BGP does not advertise the route to the CE.

EIGRP-Originated Internal Routes: If an EIGRP-originated internal route is received through BGP and the route has extended community information for EIGRP, the PE sets the route type to "internal" if the source autonomous system number matches the autonomous system number configured for this VPN routing and forwarding (VRF) instance. BGP, then, reconstructs and advertises the route to the receiving CE as an internal EIGRP route by using the extended community information. If there is no autonomous system match, these routes are treated as non-EIGRP-originated routes.

EIGRP-Originated External Routes: If an EIGRP-originated external route is received through BGP and the route has extended community information for EIGRP, the PE sets the route type to “external” if the source autonomous system number matches the autonomous system number configured for this VRF instance. BGP, then, reconstructs and advertises this external route to the receiving CE as an external EIGRP route by using the extended community information. If there is no autonomous system match, these routes are treated as non-EIGRP-originated routes.

Preserving Route Metrics

The EIGRP 6PE/6VPE feature manages native and non-native Enhanced Interior Gateway Routing Protocol (EIGRP) routes by using the **redistribute** and the **default metric** commands, respectively. By using the **redistribute bgp** *as-number* command, you can ensure that only Border Gateway Protocol (BGP) routes with BGP Extended Community information are distributed into EIGRP. EIGRP uses this information to recreate the original EIGRP route. If the BGP Extended Community information is missing and the default metric is not specified, EIGRP will not learn the route from BGP.

By using the **redistribute bgp** *as-number* **metric-type** *type-value* command, you can ensure that the metric values configured using this command are used only for BGP routes redistributed into EIGRP. EIGRP looks for BGP Extended Community information, and if this information is found, EIGRP uses this information to recreate the original EIGRP route. If the Extended Community information is missing, EIGRP uses the metric values configured using this command to determine whether the route is the preferred route.

By using the **default-metric** *bandwidth delay reliability loading mtu* command, you can ensure that the metric values configured using this command are used for any non-EIGRP routes being redistributed into EIGRP. If the received route is a BGP route, EIGRP looks for BGP Extended Community information, and if this information is found, EIGRP uses this information to recreate the original EIGRP route. If the extended community information is missing, EIGRP uses the metric values configured to determine whether the route is the preferred route.

EIGRP 6PE/6VPE SoO

The EIGRP 6PE/6VPE Site of Origin (SoO) functionality allows an Enhanced Interior Gateway Routing Protocol (EIGRP) network to support complex topologies, such as Multiprotocol Label Switching-VPN (MPLS-VPN) links between sites with backdoor links, customer-edge (CE) devices that are dual-homed to different provider-edge (PE) devices, and PEs supporting CEs from different sites within the same VPN routing and forwarding (VRF) instance. Path selection within the EIGRP network containing PE-CE links is based on route metrics that allow either the link through the VPN or the EIGRP backdoor to act as the primary (best) link or the backup link, if the primary link fails. EIGRP accomplishes this path selection by retrieving the Site of Origin (SoO) attribute from routes redistributed from the Border Gateway Protocol (BGP) network. This BGP/EIGRP interaction takes place through the use of the BGP Cost Community Extended Community attribute.

When routes are redistribued into EIGRP from a BGP network, BGP Cost Community Extended Community attributes are added to the routes. These attributes include the SoO attribute. The SoO attribute is used to identify the site of origin of a route and prevent advertisement of the route back to the source site. To enable the EIGRP SoO functionality, you must configure the **ip vrf sitemap** command on the PE interface that is connected to the CE device. This command enables SoO filtering on the interface. When EIGRP on the PE device receives CE routes on the interface that has a SoO value defined, EIGRP checks each route to determine whether there is an SoO value associated with the route that matches the interface SoO value. If the SoO values match, the route will be filtered. This filtering is done to stop routing loops.

When EIGRP on the PE receives a route that does not contain an SoO value or contains an SoO value that does not match the interface SoO value, the route will be accepted into the topology table so that it can be redistributed into BGP. When the PE redistributes an EIGRP route that does not contain an SoO value into BGP, the SoO value that is defined on the interface used to reach the next hop (CE) is included in the Extended Communities attribute associated with the route. If the EIGRP topology table entry already has an SoO value associated with the route, this SoO value, instead of the interface SoO value, will be included with the route when it is redistributed into the BGP table. Any BGP peer that receives these prefixes will also receive the SoO value associated with each prefix, identifying the site, where each prefix originated.

The EIGRP SoO functionality ensures that BGP does not follow its normal path-selection behavior, where locally derived routes (such as native EIGRP routes redistributed into BGP) are preferred over BGP-derived routes.

For more information on the Site of Origin functionality, see the "EIGRP MPLS VPN PE-CE Site of Origin" chapter in the *IP Routing: EIGRP Configuration Guide*.

Backdoor Devices

Backdoor devices are EIGRP devices that connect one EIGRP site to another, but not through the Multiprotocol Label Switching-VPN (MPLS-VPN) network. Typically, a backdoor link is used as a backup path between peering EIGRP sites if the MPLS-VPN link is down or unavailable. The metric on the backdoor link is set high enough so that the path through the backdoor will not be selected unless there is a VPN link failure. You can define Site of Origin (SoO) values on the backdoor device on interfaces connecting the device to the peering sites, thus identifying the local-site identity of the link.

When a backdoor device receives EIGRP updates or replies from a neighbor, the device checks each received route to verify that the route does not contain an SoO value that matches the ones defined on its interfaces. If the device finds a route with a SoO value that matches the value defined on any of its interfaces, the route is rejected and not included in the topology table. Typically, the reason that a route is received with a matching SoO value is that the route is learned by the other peering site through the MPLS-VPN connection and is being advertised back to the original site over the backdoor link. By filtering such routes based on the SoO value defined on the backdoor link, you can avoid short-term, invalid routing.

Additional References for EIGRP Support for 6PE/6VPE

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQs	EIGRP Frequently Asked Questions
EIGRP technology white papers	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 183: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 150

EIGRP Over the Top

The EIGRP Over the Top feature enables a single end-to-end routing domain between two or more Enhanced Interior Gateway Routing Protocol (EIGRP) sites that are connected using a private or a public WAN connection. This module provides information about the EIGRP Over the Top feature and how to configure it.

Information About EIGRP Over the Top

EIGRP Over the Top Overview

The EIGRP Over the Top feature enables a single end-to-end Enhanced Interior Gateway Routing Protocol (EIGRP) routing domain that is transparent to the underlying public or private WAN transport that is used for connecting disparate EIGRP customer sites. When an enterprise extends its connectivity across multiple sites through a private or a public WAN connection, the service provider mandates that the enterprise use an additional routing protocol, typically the Border Gateway Protocol (BGP), over the WAN links to ensure end-to-end routing. The use of an additional protocol causes additional complexities for the enterprise, such as additional routing processes and sustained interaction between EIGRP and the routing protocol to ensure connectivity, for the enterprise. With the EIGRP Over the Top feature, routing is consolidated into a single protocol (EIGRP) across the WAN, which provides the following benefits:

- There is no dependency on the type of WAN connection used.
- There is no dependency on the service provider to transfer routes.
- There is no security threat because the underlying WAN has no knowledge of enterprise routes.
- This feature simplifies dual carrier deployments and designs by eliminating the need to configure and manage EIGRP-BGP route distribution and route filtering between customer sites.
- This feature allows easy transition between different service providers.
- This feature supports both IPv4 and IPv6 environments.

How EIGRP Over the Top Works

The EIGRP Over the Top solution can be used to ensure connectivity between disparate Enhanced Interior Gateway Routing Protocol (EIGRP) sites. This feature uses EIGRP on the control plane and Locator ID Separation Protocol (LISP) encapsulation on the data plane to route traffic across the underlying WAN architecture. EIGRP is used to distribute routes between customer edge (CE) devices within the network, and the traffic forwarded across the WAN architecture is LISP encapsulated. Therefore, to connect disparate EIGRP sites, you must configure the **neighbor** command with LISP encapsulation on every CE in the network.

If your network has many CEs, then you can use EIGRP Route Reflectors (E-RRs) to form a half-mesh topology and ensure connectivity among all CEs in the network. An E-RR is an EIGRP peer that receives EIGRP route updates from CEs in the network and reflects these updates to other EIGRP CE neighbors without changing the next hop or metrics for the routes. An E-RR can also function as a CE in the network. You must configure E-RRs with the **remote-neighbors source** command to enable E-RRs to listen to unicast messages from peer CE devices and reflect the messages to other EIGRP CE neighbors. You must configure the CEs with the **neighbor** command to allow them to identify the E-RRs in their network and exchange routes with the E-RRs. Upon learning routes from E-RRs, the CEs install these routes into their routing information base (RIB). You can use dual or multiple E-RRs for redundancy. The CEs form adjacencies with all E-RRs configured in the network, thus enabling multihop remote neighborship amongst themselves.

Security Groups and SGTs

A security group is a grouping of users, endpoint devices, and resources that share access control policies. Security groups are defined by the administrator in the ACS. As new users and devices are added to the Cisco TrustSec (CTS) domain, the authentication server assigns these new entities to appropriate security groups. CTS assigns to each security group a unique 16-bit security group number whose scope is global within a CTS domain. The number of security groups in the router is limited to the number of authenticated network entities. Security group numbers do not need to be manually configured.

Once a device is authenticated, CTS tags any packet that originates from that device with an SGT that contains the security group number of the device. The packet carries this SGT throughout the network within the CTS header. The SGT is a single label that determines the privileges of the source within the entire CTS domain. The SGT is identified as the source because it contains the security group of the source. The destination device is assigned a destination group tag (DGT).

Note The CTS packet tag does not contain the security group number of the destination device.

EIGRP OTP Support to Propagate SGT

The EIGRP OTP Support enables to propagate SGT from site-to-site across WAN using OTP transport. OTP uses LISP to send the data traffic. OTP carries the SGT over the Layer 3 (L3) clouds across multiple connections/network and also provides access control at a remote site.

How to Configure EIGRP Over the Top

Configuring EIGRP Over the Top on a CE Device

You must enable the EIGRP Over the Top feature on all customer edge (CE) devices in the network so that the CEs know how to reach the Enhanced Interior Gateway Routing Protocol (EIGRP) Route Reflector configured in the network. Perform the following task to configure the EIGRP Over the Top feature on a CE device and enable Locator ID Separation Protocol (LISP) encapsulation for traffic across the underlying WAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *as-number*
5. **neighbor**{*ip-address* | *ipv6-address*} *interface-type interface-number* [**remote** *maximum-hops* [**lisp-encap** [*lisp-id*]]]
6. **network** *ip-address*[*wildcard-mask*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp test`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 100`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 5	**neighbor**{*ip-address* | *ipv6-address*} *interface-type interface-number* [**remote** *maximum-hops* [**lisp-encap** [*lisp-id*]]]	Defines a neighboring device with which an EIGRP device can exchange routing information.

	Command or Action	Purpose
	Example: `Device(config-router-af)# neighbor 10.0.0.1 gigabitethernet 0/0/1 remote 2 lisp-encap 1`	
Step 6	**network** *ip-address*[*wildcard-mask*] **Example:** `Device(config-router-af)# network 192.168.0.0 255.255.0.0`	Specifies the network for the EIGRP routing process. In this case, configure all routes that the CE needs to be aware of.
Step 7	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring EIGRP Route Reflectors

Perform this task to configure a customer edge (CE) device in a network to function as an Enhanced Interior Gateway Routing Protocol (EIGRP) Route Reflector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 unicast autonomous-system** *as-number*
5. **af-interface** *interface-type interface-number*
6. **no next-hop-self**
7. **no split-horizon**
8. **exit**
9. **remote-neighbors source** *interface-type interface-number* **unicast-listen lisp-encap**
10. **network** *ip-address*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp test`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 unicast autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv4 unicast autonomous-system 100`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 5	**af-interface** *interface-type interface-number* **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**no next-hop-self** **Example:** `Device(config-router-af-interface)# no next-hop-self`	Instructs EIGRP to use the received next hop and not the local outbound interface address as the next hop to be advertised to neighboring devices. **Note** If **no next-hop-self** is not configured, the data traffic will flow through the EIGRP Route Reflector.
Step 7	**no split-horizon** **Example:** `Device(config-router-af-interface)# no split-horizon`	Disables EIGRP split horizon.
Step 8	**exit** **Example:** `Device(config-router-af-interface)# exit`	Exits address family interface configuration mode and returns to address family configuration mode.
Step 9	**remote-neighbors source** *interface-type interface-number* **unicast-listen lisp-encap** **Example:** `Device(config-router-af)# remote-neighbors source gigabitethernet 0/0/1 unicast-listen lisp-encap`	Enables remote neighbors to accept inbound connections from any remote IP address.
Step 10	**network** *ip-address* **Example:**	Specifies a network for the EIGRP routing process.

	Command or Action	Purpose
	`Device(config-router-af)# network 192.168.0.0`	• Enter all network routes that the EIGRP Route Reflector needs to be aware of.
Step 11	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode

Configuring EIGRP OTP Support to Propagate SGT

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual instance name*
4. **address-family ipv4 autonomous-system** *as-number*
5. **topology base**
6. **cts propagate sgt**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual instance name* **Example:** `Device (config)# router eigrp kmd`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *as-number* **Example:** `Device (config-router)# address-family ipv4 autonomous-system 100`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 5	**topology base** **Example:** `Device (config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.

	Command or Action	Purpose
Step 6	**cts propagate sgt** **Example:** `Device (config-router-af)# cts propagate sgt`	Enables Security Group Tag (SGT) propagation over L3 network.
Step 7	**end** **Example:** `Device (config-router-af)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuration Examples for EIGRP Over the Top

Example: Configuring EIGRP Over the Top on a CE Device

The following example shows you how to configure the customer edge (CE) device in the network to advertise local routes to the Enhanced Interior Gateway Routing Protocol (EIGRP) Route Reflectors.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp test
Device(config-router)# address-family ipv4 unicast autonomous-system 100
Device(config-router-af)# neighbor 10.0.0.2 gigabitethernet 0/0/1 remote 3 lisp-encap 1
Device(config-router-af)# network 192.168.0.0
Device(config-router-af)# network 192.168.1.0
Device(config-router-af)# network 192.168.2.0
Device(config-router-af)# end
```

Example: Configuring EIGRP Route Reflectors

```
Device> enable
Device# configure terminal
Device(config)# router eigrp test
Device(config-router)# address-family ipv4 unicast autonomous-system 100
Device(config-router-af)# af-interface gigabitethernet 0/0/1
Device(config-router-af-interface)# no next-hop-self
Device(config-router-af-interface)# no split-horizon
Device(config-router-af-interface)# exit
Device(config-router-af)# remote-neighbors source gigabitethernet 0/0/1 unicast-listen
lisp-encap 1
Device(config-router-af)# network 192.168.0.0
Device(config-router-af)# end
```

Example: Configuring EIGRP OTP Support to Propagate SGT

The following example shows how to configure EIGRP OTP to propagate SGT.

```
router eigrp kmd
!
address-family ipv4 unicast autonomous-system 100
  !
  topology base
   cts propagate sgt
  exit-af-topology
exit-address-family
```

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 184: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 151

EIGRP OTP VRF Support

The EIGRP OTP VRF support feature extends VPN routing and forwarding (VRF) support to the EIGRP OTP feature thereby retaining and carrying VRF information over WAN.

Prerequisites for EIGRP OTP VRF Support

The EIGRP Over the Top feature must be configured.

Restrictions for EIGRP OTP VRF Support

- The WAN facing interface should not be in VRF.

Information About EIGRP OTP VRF Support

Overview of EIGRP OTP VRF Support

The EIGRP Over the Top is a WAN solution with EIGRP in control plane and LISP in data plane, in which route distribution between two EIGRP customer-edge devices is performed using EIGRP protocol. LISP encapsulates the data that is sent over WAN. To support VRF functionality, the routes from each VRF must be carried over the control plane and installed in the correct VRF tables in the CE devices and EIGRP Route Reflector (E-RR).

How EIGRP OTP VRF Support Works

A CE device supports multiple VRFs on a LAN. On a WAN, the WAN interface in the default VRF and the CE device forms a remote EIGRP neighborship with another CE or E-RR device. The neighbors are formed in a single EIGRP process. One EIGRP process handles multiple, distinct neighbor formations in various VRFs on the LAN side and at the same time, also forms a neighbor on the WAN side with an OTP peer. The receiving peer picks routes that are applicable for the topologies that are present on the receiving peer. Routes from any other topologies are dropped.

Various routes learnt from peers in different VRFs are updated in the respective topologies on the CE and are transported to the OTP peer with the topology information for each route. Each topology represents a configured VRF on the device.

Each topology is associated with a unique ID, called the TID (Topology ID). The TID identifies the topology across various remote customer sites as the VRF name could be different on each CE device. For the CE devices to exchange the right information, the TID must be the same on all CEs.

The LISP Id (LISP Instance ID) also is mapped to a VRF and TID. As LISP carries different VRF packets using different virtual LISP interfaces, the LISP ID per VRF must be unique and must be same across the CE devices for packet delivery.

Use the **topology** command to configure a unique topology ID on customer site.

Data Encapsulation

Data encapsulation is achieved using LISP and is configured using the same **topology** command. Each VRF is associated with a LISP virtual interface. Data packets from one VRF will be encapsulated between the CE devices per VRF.

Each CE device is the edge device for a customer site, having various VRFs in a network. When customer sites connect via EIGRP OTP, each CE device is a neighbor to another CE device. In case of E-RR deployment, the CE s neighbors with the E-RR. The routes in a VRF in one customer site are carried to its peer and updated in the appropriate peer VRF table. If routes are received from a particular topology is absent in a peer, the peer drops the routes.

The E-RR reflects all topologies that are configured on the E-RR. Routes from topologies that are absent on the E-RR are not reflected. This is the reason that the E-RR is expected to have a super set of all VRFs present in the network.

Interfaces and Topology Command

When the **topology** command is used, all the interfaces under that VRF are enabled with EIGRP, thereby forming neighbors on all interfaces under a VRF. However, there may be interfaces on which EIGRP should not be enabled. To disable the formation of peers on such interfaces, use the **topo-interface** command and disable the interface on which EIGRP must not be enabled via **passive-interface** command.

Differences between EIGRP OTP Feature and EIGRP OTP VRF Support Feature

Table 185: EIGRP OTP Feature and EIGRP VRF Support Feature Differences

EIGRP OTP Feature	EIGRP OTP VRF Support Feature
Supports the default VRF only.	Multiple VRFs can be configured. Each VRF is considered as a topology and the topology related information is carried across associated with a TID (topology ID).
Neighbors are formed on only those interfaces that are configured with the **network** command.	Neighbors are formed across all interfaces in a particular VRF configured with the **topology** command.
The **network** command is required on the WAN interface to form an OTP neighbor.	The **network** command is not required.

How to Configure EIGRP OTP VRF Support

Configuring EIGRP OTP VRF Support on a CE Device

You must enable the EIGRP OTP VRF Support feature on all customer edge (CE) devices in the network so that the CEs know how to reach the Enhanced Interior Gateway Routing Protocol (EIGRP) Route Reflector configured in the network. Perform the following task to configure the EIGRP OTP VRF Support feature on a CE device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *as-number*
5. **topology vrf** *vrf-name* **tid** *number* **lisp-instance-id** *number*
6. **topo-interface** *interface-name interface-number*
7. **passive-interface**
8. **exit**
9. **exit**
10. **neighbor**{*ip-address* | *ipv6-address*} *interface-type interface-number* [**remote** [**lisp-encap** [*lisp-id*]]]
11. **end**
12. **show ip eigrp topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp test`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 10`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 5	**topology vrf** *vrf-name* **tid** *number* **lisp-instance-id** *number* **Example:** `Device(config-router-af)# topology vrf vrf1 tid 10 lisp-instance-id 122`	Enters address-family topology configuration mode and assigns a topology to a VRF.
Step 6	**topo-interface** *interface-name interface-number* **Example:** `Device(config-router-af-topology)# #topo-interface GigabitEthernet0/0/0`	(Optional) Enters address family interface configuration mode and the interface on which EIGRP must not be enabled.
Step 7	**passive-interface** **Example:** `Device(config-router-af-topology-interface)# passive-interface`	Makes the interface passive.
Step 8	**exit** **Example:** `Device(config-router-af-topology-interface)# exit`	Exits address family interface configuration mode and returns to address-family topology configuration mode.
Step 9	**exit** **Example:** `Device(config-router-af-topology)# exit`	Exits address-family topology configuration mode and returns to address family configuration mode.
Step 10	**neighbor** {*ip-address* | *ipv6-address*} *interface-type interface-number* [**remote** [**lisp-encap** [*lisp-id*]]] **Example:** `Device(config-router-af)# neighbor 10.0.0.1 ATM0/3/0 remote lisp-encap 122`	Defines a neighboring device with which an EIGRP device can exchange routing information.
Step 11	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 12	**show ip eigrp topology** **Example:** `Router# show ip eigrp topology`	Displays EIGRP topology table entries.

Example

The following is a sample output from the show ip eigrp topology command.

```
Device# show ip eigrp topology

EIGRP-IPv4 VR(otp) Topology Table for AS(1)/ID(10.0.0.11)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.0.0.0/24, 1 successors, FD is 131072000
        via Connected, Ethernet0/1
EIGRP-IPv4 VR(otp) Topology Table for AS(1)/ID(10.0.0.11)
           Topology(red) TID(20) VRF(red)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 21.0.0.0/24, 1 successors, FD is 12161609142
        via 20.0.0.11 (12161609142/12096073142), Ethernet0/1
P 1.11.11.11/32, 1 successors, FD is 12161691062
        via 20.0.0.11 (12161691062/12096155062), Ethernet0/1
P 11.0.0.0/24, 1 successors, FD is 131072000
        via Connected, Ethernet0/0
P 1.1.1.1/32, 1 successors, FD is 131153920
        via 11.0.0.10 (131153920/163840), Ethernet0/0
EIGRP-IPv4 VR(otp) Topology Table for AS(1)/ID(10.0.0.11)
           Topology(green) TID(30) VRF(green)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 2.222.222.222/32, 1 successors, FD is 12161691062
        via 30.0.0.11 (12161691062/12096155062), Ethernet0/1
P 12.0.0.0/24, 1 successors, FD is 131072000
        via Connected, Ethernet0/2
P 31.0.0.0/24, 1 successors, FD is 12161609142
        via 30.0.0.11 (12161609142/12096073142), Ethernet0/1
P 11.22.11.22/32, 1 successors, FD is 12161691062
        via 30.0.0.11 (12161691062/12096155062), Ethernet0/1
P 2.2.2.2/32, 1 successors, FD is 131153920
        via 12.0.0.10 (131153920/163840), Ethernet0/2
P 22.0.0.0/24, 1 successors, FD is 12161609142
        via 20.0.0.11 (12161609142/12096073142), Ethernet0/1
P 2.22.22.22/32, 1 successors, FD is 12161691062
        via 20.0.0.11 (12161691062/12096155062), Ethernet0/1
EIGRP-IPv4 VR(otp) Topology Table for AS(1)/ID(10.0.0.11)
           Topology(blue) TID(40) VRF(blue)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 13.0.0.0/24, 1 successors, FD is 131072000
        via Connected, Ethernet0/3
P 32.0.0.0/24, 1 successors, FD is 12161609142
        via 30.0.0.11 (12161609142/12096073142), Ethernet0/1
P 3.33.33.33/32, 1 successors, FD is 12161691062
        via 30.0.0.11 (12161691062/12096155062), Ethernet0/1
P 3.3.3.3/32, 1 successors, FD is 131153920
        via 13.0.0.10 (131153920/163840), Ethernet0/3
```

Configuring EIGRP OTP VRF Support on EIGRP Route Reflectors

Perform this task to configure a customer edge (CE) device in a network to function as an Enhanced Interior Gateway Routing Protocol (EIGRP) Route Reflector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *as-number*
5. **topology vrf** *vrf-name* **tid** *number* **lisp-instance-id** *number*
6. **exit**
7. **remote-neighbors source** *interface-type interface-number* **unicast-listen lisp-encap** *LISP-instance-ID*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp test`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv4 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 10`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 5	**topology vrf** *vrf-name* **tid** *number* **lisp-instance-id** *number* **Example:** `Device((config-router-af)# topology vrf vrf1 tid 10 lisp-instance-id 122`	Assigns a topology to a VRF and enters address-family topology configuration mode.
Step 6	**exit** **Example:**	Exits address-family topology configuration mode and returns to address family configuration mode.

	Command or Action	Purpose
	`Device((config-router-af-topology)# exit`	
Step 7	**remote-neighbors source** *interface-type interface-number* **unicast-listen lisp-encap** *LISP-instance-ID* **Example:** `Device(config-router-af)# remote-neighbors source ATM0/3/0 unicast-listen lisp-encap 122`	Enables remote neighbors to accept inbound connections from any remote IP address.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for EIGRP OTP VRF Support

Example: Configuring EIGRP OTP VRF Support on a CE Device

```
Router> enable
Router# configure terminal
Device(config)# router eigrp test
Device(config-router)# address-family ipv4 autonomous-system 10
Device((config-router-af)# topology vrf vrf1 tid 10 lisp-instance-id 122
Device(config-router-af-topology)# topo-interface GigabitEthernet0/0/0
Device(config-router-af-topology-interface)# passive-interface
Device(config-router-af-topology-interface)# exit
Device((config-router-af-topology)# exit
Device(config-router-af)# neighbor 10.0.0.1 ATM0/3/0 remote lisp-encap 122
Device(config-router-af)# end
```

Example: Configuring EIGRP OTP VRF Support on EIGRP Route Reflectors

```
Device> enable
Device# configure terminal
Device(config)# router eigrp test
Device(config-router)# address-family ipv4 autonomous-system 10
Device(config-router-af)# topology vrf vrf1 tid 10 lisp-instance-id 122
Device(config-router-af-topology)# exit
Device(config-router-af)# remote-neighbors source ATM0/3/0 unicast-listen lisp-encap 122
Device(config-router-af)# end
```

Additional References for EIGRP OTP VRF Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP Routing: EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 186: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 152

EIGRP Classic to Named Mode Conversion

The EIGRP Classic to Named Mode Conversion feature allows you to upgrade Enhanced Interior Gateway Routing Protocol (EIGRP) classic mode configurations to named mode configurations without causing network flaps or requiring the EIGRP process to restart. This feature supports both IPv4 and IPv6.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for EIGRP Classic to Named Mode Conversions

- You must use the **eigrp upgrade-cli** command to convert Enhanced Interior Gateway Routing Protocol (EIGRP) configurations from classic mode to named mode. If multiple classic mode configurations exist, you must use this command per EIGRP autonomous system number in classic mode.
- The **eigrp upgrade-cli** command blocks the router from accepting any other command until the conversion is complete (the console is locked). The time taken to complete the conversion depends on the size of the configuration. However, the conversion is a one-time activity.
- The **eigrp upgrade-cli** command is available only under EIGRP classic router configuration mode. Therefore, you can convert configurations from classic mode to named mode but not vice-versa.
- After conversion, the running configuration on the device will show only named mode configurations; you will be unable to see any classic mode configurations. To revert to classic mode configurations, you can reload the router without saving the running configuration to the startup configuration.

- The new configurations are available only in the running configuration; they will not be saved to the startup configuration. If you want to add them to the startup configuration, you must explicitly save them using the **write memory** or the **copy running-config startup-config** command.
- After conversion, the **copy startup-config running-config** command will fail because you cannot have both the classic and named mode for the same autonomous system.
- After conversion, all neighbors (under the converted router EIGRP) will undergo graceful restart and sync all routes.

Information About EIGRP Classic to Named Mode Conversion

-

EIGRP Classic to Named Mode Conversion - Overview

The Enhanced Interior Gateway Routing Protocol (EIGRP) can be configured using either the classic mode or the named mode. The classic mode is the old way of configuring EIGRP. In classic mode, EIGRP configurations are scattered across the router mode and the interface mode. The named mode is the new way of configuring EIGRP; this mode allows EIGRP configurations to be entered in a hierarchical manner under the router mode.

Each named mode configuration can have multiple address families and autonomous system number combinations. In the named mode, you can have similar configurations across IPv4 and IPv6. We recommend that you upgrade to EIGRP named mode because all new features, such as Wide Metrics, IPv6 VRF Lite, and EIGRP Route Tag Enhancements, are available only in EIGRP named mode.

Use the **eigrp upgrade-cli** command to upgrade from classic mode to named mode. You must use the **eigrp upgrade-cli** command for all classic router configurations to ensure that these configurations are upgraded to the named mode. Therefore, if multiple classic configurations exist, you must use this command per autonomous system number. You must use this command separately for IPv4 and IPv6 configurations.

Prior to the EIGRP Classic to Named Mode Conversion feature, upgrading to EIGRP named mode required that the user manually unconfigure the classic mode using the **no router eigrp** *autonomous-system-number* command and then reconfigure EIGRP configurations under named mode using the **router eigrp** *virtual name* command. This method may lead to network churn and neighborship or network flaps.

The EIGRP Classic to Named Mode Conversion feature allows you to convert from classic mode to named mode without causing network flaps or the EIGRP process to restart. With this feature, you can move an entire classic mode configuration to a router named mode configuration, and consequently, all configurations under interfaces will be moved to the address-family interface under the appropriate address family and autonomous-system number. After conversion, the **show running-config** command will show only named mode configurations; you will not see any old classic mode configurations.

Additional References for EIGRP Classic to Named Mode

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP technology white paper	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 187: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 153

EIGRP Scale for DMVPN

The EIGRP Scale for DMVPN feature provides an increase in hub scalability for Dynamic Multipoint VPN (DMVPN). Cisco DMVPN is a security solution for building scalable enterprise VPNs that support distributed applications such as voice and video.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About EIGRP Scale for DMVPN

EIGRP Scale for DMVPN Overview

Dynamic Multipoint VPN (DMVPN) improves the usage of spoke-to-spoke networks. However, scaling of routing protocols and optimization of routing updates in large scale DMVPN networks remain a challenge. These challenges pertain to neighbor discovery, overhead reduction, and building upon the recent enhancements in the area of scaling routing over DMVPN. IPSEC tunnels, Next Hop Resolution Protocol (NHRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) are established during initial startup of a DMVPN network. It is possible that EIGRP may not process and respond to inbound packets waiting in the interface or socket queue causing the spokes to time out and retransmit which worsens the resource contention issue. The EIGRP Scale for DMVPN feature provides an increase in the scalability of the hub device to 2500 sessions. The increase in the number of sessions reduces the adverse impact on CPU, system buffers, interface buffers, and queues and it reduces resource contention on the hub during initial startup of a DMVPN network. In a typical EIGRP DMVPN setup, spokes are configured as stubs.

This EIGRP Scale for DMVPN feature is enabled by default and does not have a configuration task.

Additional References for EIGRP Scale for DMVPN

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 188: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

CHAPTER 154

EIGRP IWAN Simplification

EIGRP is widely deployed on DMVPN networks. The EIGRP IWAN Simplification feature implements stub site behavior for EIGRP deployed on DMVPN networks.

Information About EIGRP IWAN Simplification

Stub Site ID Configuration

The EIGRP Stub Routing feature improves network stability, reduces resource utilization, and simplifies the stub device configuration on the spoke. EIGRP Stub routing is commonly used over DMVPN networks having multiple sites with single device in each site. Site devices acting as stub result in reducing the query domain thereby enhancing improved performance. On the other hand, branch EIGRP routing is simple for a single router default-gateway site. When a the branch adds a second router or becomes larger and needs routing within the campus the configuration becomes complex.

The EIGRP IWAN Simplification feature implements stub site behavior on devices that are connected to the WAN interfaces on branch routing via the configuration of stub site ID on EIGRP address family. Use the **eigrp stub-site** command in the address family configuration mode. The stub site ID is applied to all incoming routes on WAN interfaces.

Note The **eigrp stub-site** command is mutually exclusive with the **eigrp stub** command. You cannot execute both commands on a device. This **eigrp stub-site** command resets the peers on WAN interfaces and initiates relearning of routes from WAN neighbors.

Interfaces connected towards hub or WAN are identified so that routes learnt through neighbors on such interfaces are part of a list of a given route. This is achieved via the **stub-site wan-interface** command configured in the address family interface configuration mode.

Note On the identified interfaces, neighbors treat WAN interfaces as stub.

How to Configure EIGRP IWAN Simplification

Configuring the Stub Site ID

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [**wildcard-mask**]
6. **stub-site wan-interface**
7. **end**
8. **show ip eigrp vrf** *vrf-name* **topology** [*ip-address* [*mask*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

	Command or Action	Purpose

	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**network** *ip-address* [**wildcard-mask**] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution device.
Step 6	**stub-site wan-interface** **Example:** `Device(config-router-af-interface)# stub-site wan-interface`	Specifies a stub site for the WAN interfaces.
Step 7	**end** **Example:** `Device(config-router-af-interface)# end`	Exits the address family interface configuration mode and returns to privileged EXEC mode.
Step 8	**show ip eigrp vrf** *vrf-name* **topology** [*ip-address* [*mask*]] **Example:** `Device# show ip eigrp vrf vrf1 topology 109.1.0.6/32`	Displays VPN routing and forwarding (VRF) entries in the EIGRP topology table.

Example

The following is a sample output from the **show ip eigrp vrf topology** command

```
Device# show ip eigrp vrf vrf1 topology 109.1.0.6/32

EIGRP-IPv4 Topology Entry for AS(1)/ID(109.1.0.2) VRF(vrf1)
EIGRP-IPv4(1): Topology base(0) entry for 109.1.0.6/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2713600
  Descriptor Blocks:
  104.1.1.58 (Tunnel1), from 104.1.1.1, Send flag is 0x0
      Composite metric is (2713600/1408256), route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
       Total delay is 105000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 200
        Hop count is 2
        Originating router is 109.1.0.6
      Extended Community: StubSite:101:100
```

Configuration Examples for EIGRP IWAN Simplification

Example: Configuring the Stub Site ID

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
Device(config-router-af)# eigrp stub-site 101:100
Device(config-router-af)# af-interface gigabitethernet 0/0/1
Device(config-router-af-interface)# stub-site wan-interface
Device(config-router-af-interface)# end
```

Additional References for EIGRP IWAN Simplification

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of Cisco TrustSec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 189: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
IPv6 enablement - Inline Tagging	Cisco IOS XE Fuji 16.8.1	The support for IPv6 is introduced.

PART V

ISIS

CHAPTER 155

IS-IS Overview and Basic Configuration

This module provides a technical overview of the Integrated Intermediate System-to-Intermediate System (IS-IS) routing protocol. IS-IS is a link-state Interior Gateway Protocol (IGP). Link-state protocols are characterized by the propagation of the information required to build a complete network connectivity map on each participating device. That map is then used to calculate the shortest path to destinations.

The IS-IS protocol was developed in the late 1980s by Digital Equipment Corporation (DEC) and was standardized by the International Standards Organization (ISO) in ISO/IEC 10589. The current version of this standard is ISO/IEC 10589:2002.

ISO/IEC 10589 defines support for the ISO Connectionless Network Protocol (CLNP) as defined in ISO 8473. However, the protocol was designed to be extensible to other network protocols. RFC 1195 defined IS-IS support for IP, and additional IETF extensions have defined IS-IS support for IPv6. Integration of support for multiple network layer protocols has led to the term Integrated IS-IS. The Cisco IOS IS-IS implementation supports CLNP, IPv4, and IPv6. This module and its related modules use the term IS-IS to refer to the Integrated IS-IS that is implemented by Cisco IOS software.

Prerequisites for IS-IS Overview and Basic Configuration

- This document assumes knowledge of CLNS, IPv4, and IPv6.
- The amount of knowledge required for each technology is dependent on your deployment. You should know your network design and how you want traffic to flow through it before configuring IS-IS.
- Define areas, prepare an addressing plan for the devices (including defining the NETs), and determine the interfaces that will run Integrated IS-IS.
- To facilitate verification, a matrix of adjacencies should be prepared before you configure your devices, showing what neighbors should be expected in the adjacencies table.

Information About IS-IS Overview and Basic Configuration

IS-IS Functional Overview

A routing domain may be divided into one or more subdomains. Each subdomain is referred to as an area and is assigned an area address. Routing within an area is referred to as Level-1 routing. Routing between Level-1 areas is referred to as Level-2 routing. A device in Open Systems Interconnection (OSI) terminology is referred to as an Intermediate System (IS). An IS may operate at Level 1, Level 2, or both. ISs that operate at Level 1 exchange routing information with other Level-1 ISs in the same area. ISs that operate at Level 2 exchange routing information with other Level-2 devices regardless of whether they are in the same Level-1 area. The set of Level-2 devices and the links that interconnect them form the Level-2 subdomain, which must not be partitioned in order for routing to work properly.

IS Address Assignment

An IS is identified by an address known as a Network Entity Title (NET). The NET is the address of a Network Service Access Point (NSAP), which identifies an instance of the IS-IS routing protocol running on an IS. The NET may be 8 to 20 octets in length and consists of three parts:

- Area address—This field is 1 to 13 octets in length and is composed of high-order octets of the address.

Note An IS-IS instance may be assigned multiple area addresses. When this is the case, all area addresses are considered synonymous. Multiple synonymous area addresses are useful when merging or splitting areas in the domain. In normal operation, for example, once the merge or split has been completed, there is no need to assign more than one area address to an IS-IS instance.

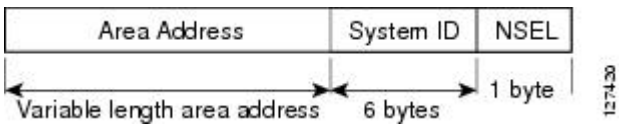
- System ID—This field is 6 octets long and immediately follows the area address. When the IS operates at Level 1, the system ID must be unique among all the Level-1 devices in the same area. When the IS operates at Level 2, the system ID must be unique among all devices in the domain.

Note An IS instance is assigned exactly one system ID.

- NSEL—The N-selector field is 1 octet in length and immediately follows the system ID. It must be set to 00.

The figure below shows the format for the NET.

Figure 140: NET Format



IS-IS PDU Types

ISs exchange routing information with their peers using protocol data units (PDUs). The following types of PDUs are used:

IIHs

Intermediate System-to-Intermediate System Hello PDUs (IIHs) are exchanged between IS neighbors on circuits on which the IS-IS protocol is enabled. IIHs include the system ID of the sender, the assigned area address(es), and the identity of neighbors on that circuit that are known to the sending IS. Additional optional information may also be included.

There are three types of IIHs:

- Point-to-Point IIHs—These are sent on point-to-point circuits.
- Level-1 LAN IIHs—These are sent on multiaccess circuits when the sending IS operates as a Level-1 device on that circuit.
- Level-2 LAN IIHs—These are sent on multiaccess circuits when the sending IS operates as a Level-2 device on that circuit.

LSPs

An IS generates Link-State PDUs (LSPs) to advertise its neighbors and the destination that are directly connected to the IS. An LSP is uniquely identified by the following:

- System ID of the IS that generated the LSP
- Pseudonode ID—This value is always 0 except when the LSP is a pseudonode LSP (see “Operation of IS-IS on Multiaccess Circuits” section.
- LSP number (0 to 255)
- 32-bit sequence number

Whenever a new version of an LSP is generated, the sequence number is incremented.

Level-1 LSPs are generated by ISs that support Level 1. The Level-1 LSPs are flooded throughout the Level-1 area. The set of Level-1 LSPs generated by all Level-1 ISs in an area is the Level-1 LSP Database (LSPDB). All Level-1 ISs in an area will have an identical Level-1 LSPDB and will therefore have an identical network connectivity map for the area.

Level-2 LSPs are generated by ISs that support Level 2. Level-2 LSPs are flooded throughout the Level-2 subdomain. The set of Level-2 LSPs generated by all Level-2 ISs in the domain is the Level-2 LSP Database (LSPDB). All Level-2 ISs will have an identical Level-2 LSPDB and will therefore have an identical connectivity map for the Level-2 subdomain.

SNPs

Sequence Number PDUs (SNPs) contain a summary description of one or more LSPs. There are two types of SNPs for both Level 1 and Level 2:

- Complete Sequence Number PDUs (CSNPs) are used to send a summary of the LSPDB that an IS has for a given level.

- Partial Sequence Number PDUs (PSNPs) are used to send a summary of a subset of the LSPs for a given level that an IS either has in its database or needs to obtain.

For more information about how SNPs are used, see the "IS-IS Supported Circuit Types" section.

IS-IS Supported Circuit Types

IS-IS supports two generic circuit types:

- Point-to-point circuits
- Multiaccess circuits

Operation of IS-IS on Point-to-Point Circuits

A point-to-point circuit has exactly two ISs on the circuit. An IS forms a single adjacency to the other IS on the point-to-point circuit. The adjacency type describes what level(s) are supported on that circuit.

If both ISs support Level 1 on that circuit and the ISs are configured with at least one matching address, the adjacency supports Level 1. Level-1 LSPs and SNPs will be sent on that circuit.

If both ISs support Level 2 on that circuit, the adjacency supports Level 2. Level-2 LSPs and SNPs will be sent on that circuit.

The adjacency then can be Level 1, Level 2, or Level 1 and 2.

ISs send point-to-point IIHs on point-to-point circuits. These IIHs allow each IS to discover the identity of the neighbor, the configured area address(es), and the supported levels.

When an adjacency is first established, each IS sends a set of CSNPs for each level that is supported on the circuit. A CSNP set describes the current contents of the LSPDB at that level. By comparing the contents of the set of received CSNPs with the contents of the local LSPDB, each IS can determine where the databases differ and initiate procedures to exchange the necessary LSPs so that the databases are efficiently and reliably synchronized.

PSNPs are sent to acknowledge the receipt of an updated LSP.

Operation of IS-IS on Multiaccess Circuits

Multiaccess circuits support multiple ISs; for example, two or more operating on the circuit. The ability to address multiple systems utilizing a multicast or broadcast address is assumed.

An IS that supports Level 1 on a multiaccess circuit sends Level-1 LAN IIHs on the circuit. An IS that supports Level 2 on a multiaccess circuit sends Level-2 LAN IIHs on the circuit.

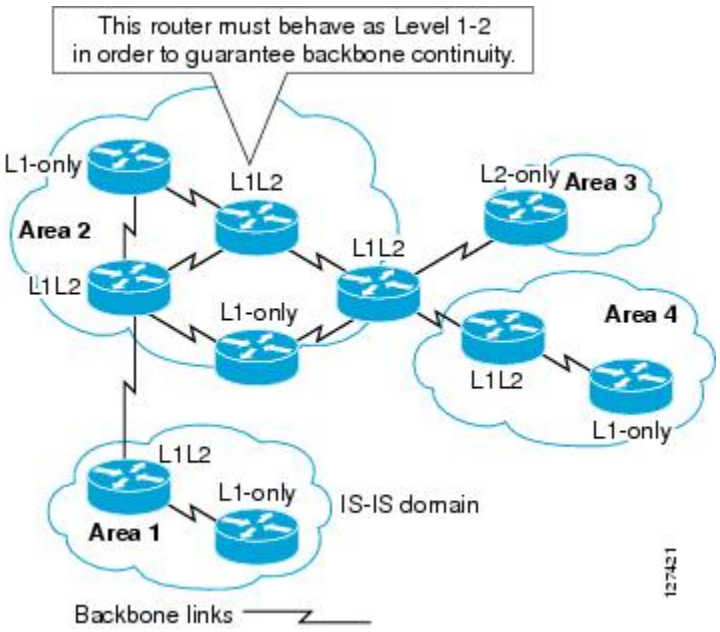
ISs form separate adjacencies for each level with neighbor ISs on the circuit.

An IS will form a Level-1 adjacency with other ISs that support Level 1 on the circuit and will have a matching area address. It is a misconfiguration to have two ISs with disjoint sets of area addresses supporting Level 1 on the same multiaccess circuit.

An IS will form a Level-2 adjacency with other ISs that support Level 2 on the circuit.

The devices in the IS-IS network topology in the figure below perform Level 1, Level 2, or Level 1 and 2 routing along the backbone of the network.

Figure 141: Level 1, Level 2, and Level 1-2 Devices in an IS-IS Network Topology



IS-IS Election of the Designated Intermediate System

If each IS advertised all of its adjacencies on a multiaccess circuit in its LSPs, the total number of advertisements required would be N 2—where N is the number of ISs that operate at a given level on the circuit. To address this scalability issue, IS-IS defines a pseudonode to represent the multiaccess circuit. All ISs that operate on the circuit at a given level elect one of the ISs to act as the Designated Intermediate System (DIS) on that circuit. A DIS is elected for each level that is active on the circuit.

The DIS is responsible for issuing pseudonode LSPs. The pseudonode LSPs include neighbor advertisements for all of the ISs that operate on that circuit. All ISs that operate on the circuit (including the DIS) provide a neighbor advertisement to the pseudonode in their non-pseudonode LSPs and do not advertise any of their neighbors on the multiaccess circuit. In this way the total number of advertisements required varies as a function of N—the number of ISs that operate on the circuit.

A pseudonode LSP is uniquely classified by the following identifiers:

- System ID of the DIS that generated the LSP
- pseudonode ID—ALWAYS NON-ZERO
- LSP number (0 to 255)
- 32-bit sequence number

The nonzero pseudonode ID is what differentiates a pseudonode LSP from a nonpseudonode LSP and is chosen by the DIS to be unique among any other LAN circuits for which it is also the DIS at this level.

The DIS is also responsible for sending periodic CSNPs on the circuit. This provides a complete summary description of the current contents of the LSPDB on the DIS. Other ISs on the circuit can then perform the following activities:

- Flood LSPs that they have that are absent from or are newer than those that are described in the CSNPs sent by the DIS.
- Request an LSP by sending a PSNP for LSPs that are described in the CSNPs sent by the DIS that are absent from the local database or older than what is described in the CSNP set.

In this way, the LSPDBs of all ISs on a multiaccess circuit are efficiently and reliably synchronized.

IS-IS Overview of LSPDB Synchronization

Proper operation of IS-IS requires a reliable and efficient process to synchronize the LSPDBs on each IS. In IS-IS this process is called the update process. This section provides a brief overview of the operation of the update process. The update process operates independently at each supported level.

LSPs may be locally generated, in which case they always are new LSPs. LSPs may also be received from a neighbor on a circuit, in which case they may be generated by some other IS or may be a copy of an LSP generated by the local IS. Received LSPs may be older, the same age, or newer than the current contents of the local LSPDB.

Handling of Newer LSPs

A newer LSP is added to the local LSPDB. If an older copy of the same LSP currently exists in the LSPDB, it is replaced. The newer LSP is marked to be sent on all circuits on which the IS currently has an adjacency in the UP state at the level associated with the newer LSP—excluding the circuit on which the newer LSP was received.

On point-to-point circuits, the newer LSP will be flooded periodically until the neighbor acknowledges its receipt by sending a PSNP or by sending an LSP that is the same or newer than the LSP being flooded.

On multiaccess circuits, the IS will flood the newer LSP once. The IS examines the set of CNSPs that are sent periodically by the DIS for the multiaccess circuit. If the local LSPDB contains one or more LSPs that are newer than what is described in the CSNP set (this includes LSPs that are absent from the CSNP set) those LSPs are reflooded over the multiaccess circuit. If the local LSPDB contains one or more LSPs that are older than what is described in the CSNP set (this includes LSPs described in the CSNP set that are absent from the local LSPDB), a PSNP is sent on the multiaccess circuit with descriptions of the LSPs that require updating. The DIS for the multiaccess circuit responds by sending the requested LSPs.

Handling of Older LSPs

An IS may receive an LSP that is older than the copy in the local LSPDB. An IS may receive an SNP (complete or partial) that describes an LSP that is older than the copy in the local LSPDB. In both cases the IS marks the LSP in the local database to be flooded on the circuit on which the older LSP or SNP that contained the older LSP was received.

At this point, the actions taken are identical to the actions that are described in the “Handling of Newer LSPs” section after a new LSP has been added to the local database.

Handling LSPs That Are the Same

Because of the distributed nature of the update process, it is possible than an IS may receive copies of an LSP that is the same as the current contents of the local LSPDB.

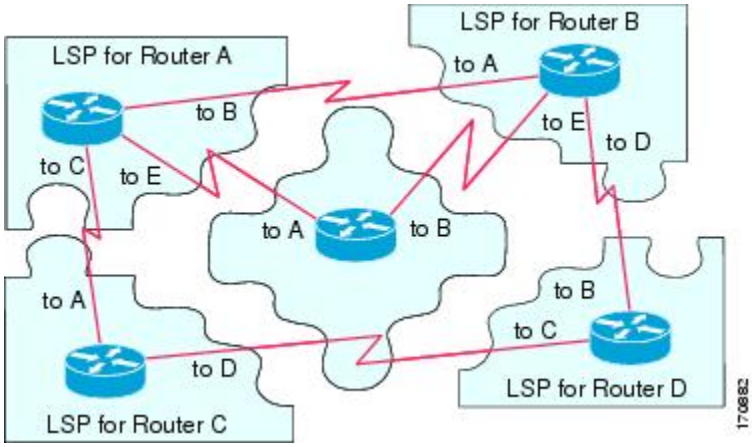
On a point-to-point circuit, receipt of such an LSP is ignored. Periodic transmission of a CSNP set by the DIS for that circuit will serve as an implicit acknowledgement to the sender that the LSP has been received.

In a multiaccess circuit, receipt of such an LSP is ignored. Periodic transmission of a CSNP set by the DIS for that circuit will serve as an implicit acknowledgement to the sender that the LSP has been received.

The figure below shows how the LSPs are used to create a network map. Imagine the network topology as a jigsaw puzzle. Each LSP (representing an IS) is considered one of the jigsaw pieces.

Note The figure below is applicable to all Level-1 devices in an area or to all Level-2 devices in a Level-2 subdomain.

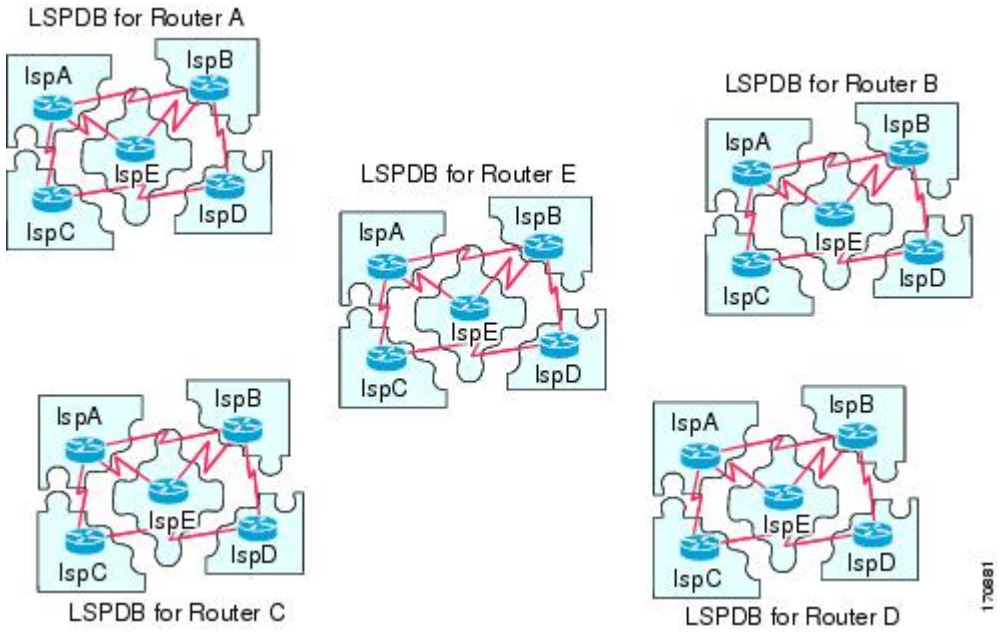
Figure 142: IS-IS Network Map



The figure below shows each device in the IS-IS network with its fully updated link-state database, after the adjacencies have been formed among the neighbor devices.

Note The figure below is applicable to all Level-1 devices in an area or to all Level-2 devices in a Level-2 subdomain.

Figure 143: IS-IS Devices with Synchronized LSPDBs



IS-IS Overview of the Shortest Path Calculation

When the contents of the LSPDB change, each IS independently reruns a shortest path calculation. The algorithm is based on the well-known Dijkstra algorithm for finding the shortest paths along a directed graph where the ISs are the vertices of the graph and the links between the ISs are edges with a nonnegative weight. A two-way connectivity check is performed before considering a link between two ISs as part of the graph. This prevents the use of stale information in the LSPDB, for example, when one IS is no longer operating in the network but did not purge the set of LSPs that it generated before ceasing operation.

The output of the SPF is a set of tuples (destination, next hop). The destinations are protocol-specific; for example, they would be prefixes when the supported protocol is IP, NSAPs of end systems when the supported protocol is CLNP. Multiple equal-cost paths are supported, in which case multiple next hops would be associated with the same destination.

Independent SPFs are performed for each level supported by the IS. In cases in which the same destination is reachable by both Level-1 and Level-2 paths, the Level-1 path is preferred.

A Level-2 IS that indicates that it has one or more Level-2 neighbors in other areas may be used by Level-1 devices in the same area as the path of last resort, also called the default route. The Level-2 IS indicates its attachment to other areas by setting an attached bit (ATT) in its Level-1 LSP 0.

Note An IS can generate up to 256 LSPs at each level. The LSPs are identified by the numbers 0 through 255. LSP 0 has special properties, including the significance of the setting of the ATT bit to indicate attachment to other areas. When LSPs that are numbered 1 through 255 have the ATT bit set, it is not significant.

How to Create Monitor and Make Changes to a Basic IS-IS Network

Enabling IS-IS as an IP Routing Protocol on the Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **net** *network-entity-title*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Assigns a tag to an IS-IS process. Enters router configuration mode. • Configure tags to identify multiple IS-IS processes by giving a meaningful name for each routing process. If the tag is not specified, a null tag (0) is assumed and the process is referenced with a null tag. The tag name must be unique among all IP router processes for the device.
Step 4	**net** *network-entity-title* **Example:** `Device(config-router)# net 49.0001.0000.0000.000b.00`	Configures the NET on the device. • The NET identifies the device for IS-IS.
Step 5	**end** **Example:**	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router)# end`	

Enabling IS-IS as an IP Routing Protocol on the Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [*secondary*]
5. **ip router isis [***area-tag***]**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:** `Device(config-if)# ip address 172.16.1.27 255.255.255.0`	Sets the primary IP address on the interface.
Step 5	**ip router isis [***area-tag***]** **Example:** `Device(config-if)# ip router isis company1`	Enables IS-IS on the interfaces that are to use IS-IS to distribute their IP information (and additionally that might be used to establish IS-IS adjacencies). • Use the *area-tag* argument to specify to which IS-IS process the device belongs. • If there is more than one IS-IS process on the device, repeat the **ip router isis** command for each interface, specifying an area tag for each interface to associate each interface with the specific process to which it belongs.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring IS-IS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isis display delimiter** [**return** *count* | **character** *count*]
4. **exit**
5. **show ip protocols**
6. **show clns** *area-tag* **is-neighbors** [*type number*] [**detail**]
7. **show clns interface** [*type number*]
8. **show clns** *area-tag* **neighbors** [*type number*] [**area**] [**detail**]
9. **show clns** *area-tag* **traffic**
10. **show ip route** [*ip-address* [*mask*]] [[**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**]]
11. **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
12. **show isis database verbose**
13. **show isis lsp-log**
14. **show isis** [*area-tag*] [**ipv6** | *****] **spf-log**
15. **show isis** [*process-tag*] [**ipv6** | *****] **topology**
16. **show isis** [*area-tag*] **neighbors** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**isis display delimiter** [**return** *count*	**character** *count*] **Example:** `Device(config)# isis display delimiter return 3`	Makes output from multiarea displays easier to read by specifying the delimiter to use to separate displays of information.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip protocols** **Example:** `Device# show ip protocols`	Displays the parameters and current state of the active routing protocol process. • You can use this command to learn what protocols are active, what interfaces they are active on, what networks they are routing for, and other parameters that relate to the routing protocols.
Step 6	**show clns** *area-tag* **is-neighbors** [*type number*] [**detail**] **Example:** `Device# show clns is-neighbors detail`	Displays IS-IS information for IS-IS device adjacencies.
Step 7	**show clns interface** [*type number*] **Example:** `Device# show clns interface`	List the CLNS-specific information about each interface.
Step 8	**show clns** *area-tag* **neighbors** [*type number*] [**area**] [**detail**] **Example:** `Device# show clns area3 neighbors`	Displays both ES and IS neighbors. • The **show clns neighbor** command output verifies that the right adjacencies have established. A matrix of adjacencies should be prepared before you configure your devices, showing what neighbors should be expected in the adjacencies table, to facilitate verification.
Step 9	**show clns** *area-tag* **traffic** **Example:** `Device# show clns area3 traffic`	Displays traffic statistics. To monitor IS-IS for stability once it has been deployed across your network, enter the **show clns traffic** command to check the following important statistics: high numbers of SPFs, checksum errors, and retransmissions. To troubleshoot IS-IS behavior, you can use the output from the **show clns traffic** command to check for the following indicators: • The number of link-state PDUs (LSPs) can help you determine the stability of the IS-IS network. The number of LSPs should never be zero. However, an LSP count that keeps increasing over a short time period indicates a network issue. • LSP retransmissions should stay low. A later execution of the **show clns traffic** command that shows an increase in LSP retransmissions, as

	Command or Action	Purpose
		compared to an earlier execution of the command, can indicate instability or traffic problems. • To check for partial route calculations (PRCs), enter the **show clns traffic** command. PRCs are flooded when a change that does not affect topology is reported through an LSP; typical examples include the addition or removal of a prefix or metric changes for external or passive interfaces. A PRC update queue that remains full or increases to the maximum value for long periods of time indicates network instability. • LSP checksum errors indicate a problem. • The update queue should not stay full and should not drop much.
Step 10	**show ip route** [*ip-address* [*mask*]] [[**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**]] **Example:** `Device# show ip route 172.16.0.21`	Displays the current state of the routing table.
Step 11	**show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:** `Device# show isis database detail`	Displays additional information about the IS-IS database. • Displays the link-state database for Level-1 and Level-2, the contents for each LSP, and the link-state protocol PDU identifier.
Step 12	**show isis database verbose** **Example:** `Device# show isis database verbose`	Displays additional information about the IS-IS database such as the sequence number, checksum, and holdtime for LSPs.
Step 13	**show isis lsp-log** **Example:** `Device# show isis lsp-log`	Displays a log of LSPs including time of occurrence, count, interface, and the event that triggered the LSP.
Step 14	**show isis** [*area-tag*] [**ipv6** | *****] **spf-log** **Example:** `Device# show isis spf-log`	Displays how often and why the device has run a full shortest path first (SPF) calculation. • If the device continues to run SPF without ceasing, there might be an issue regarding a change in the network (intra-area). The cause for the continued SPF calculations could be an interconnecting link that is transitioning up/down/up/down or a metric change. It is normal for the SPF calculation to run a few times

	Command or Action	Purpose
		when a network change occurs, but then it should cease.
Step 15	**show isis** [*process-tag*] [**ipv6** | *****] **topology** **Example:** `Device# show isis topology`	Displays a list of all connected devices in all areas.
Step 16	**show isis** [*area-tag*] **neighbors** [**detail**] **Example:** `Device# show isis neighbors detail`	Displays IS-IS adjacency information. • The **show isis neighbor detail** command output verifies that the right adjacencies have established. A matrix of adjacencies should be prepared before you configure your devices, showing what neighbors should be expected in the adjacencies table, to facilitate verification.

Example

When the **show isis neighbors** command is entered with the **detail** keyword, the output provides information about the IS-IS adjacencies that have formed.

```
Device1# show isis neighbors detail

System Id      Type Interface IP Address      State Holdtime Circuit Id
Device2         L2   Et1/0     10.1.1.0        UP   255      Circuit3.01
  Area Address(es): 32
  SNPA: aabb.cc00.2001
  State Changed: 00:00:14
  LAN Priority: 64
  Format: Phase V
```

Troubleshooting Tips

You can use the following two system debugging commands to check your IS-IS IPv4 implementation.

- If adjacencies are not coming up properly, use the **debug isis adj-packets** command.
- To display a log of significant events during an IS-IS SPF calculation, use the **debug isis spf-events** command.

Configuration Examples for a Basic IS-IS Network

Example: Configuring a Basic IS-IS Network

The following example shows how to configure three devices to run IS-IS as an IP routing protocol.

Device A Configuration

```
router isis
 net 49.0001.0000.0000.000a.00
interface ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
interface serial 2/0
 ip router isis
 ip address 192.168.1.2 255.255.255.0
```

Device B Configuration

```
router isis
 net 49.0001.0000.0000.000b.00
interface ethernet0/0
 ip router isis
 ip address 172.17.1.1 255.255.255.0
interface serial2/0
 ip router isis
 ip address 192.168.1.1 255.255.255.0
interface serial5/0
 ip router isis
 ip address 172.21.1.1 255.255.255.0
```

Device C Configuration

```
router isis
 net 49.0001.0000.0000.000c.00
interface ethernet2/0
 ip router isis
 ip address 172.21.1.2 255.255.255.0
interface serial5/0
 ip router isis
 ip address 172.22.1.1 255.255.255.0
```

The **show isis topology** command displays the following information about how the devices are connected within the IS-IS network:

```
DeviceB# show isis topology

IS-IS paths to level-1 routers
System Id            Metric     Next-Hop             Interface   SNPA
DeviceA              10         DeviceA              Se2/0       *HDLC*
DeviceB              --
DeviceC              10         DeviceC              Se5/0       *HDLC*
IS-IS paths to level-2 routers
System Id            Metric     Next-Hop             Interface   SNPA
DeviceA              10         DeviceA              Se2/0       *HDLC*
DeviceB              --
DeviceC              10         DeviceC              Se5/0       *HDLC*
```

The **show isis database** command displays following information for the Level 1 and Level 2 LSPs for each device in the IS-IS network.

```
DeviceB# show isis database

IS-IS Level-1 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
```

```
DeviceA.00-00         0x00000005   0x1A1D        1063              0/0/0
DeviceB.00-00       * 0x00000006   0xD15B        1118              0/0/0
DeviceC.00-00         0x00000004   0x3196        1133              1/0/0
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
DeviceA.00-00         0x00000008   0x0BF4        1136              0/0/0
DeviceB.00-00       * 0x00000008   0x1701        1137              0/0/0
DeviceC.00-00         0x00000004   0x3624        1133              0/0/0
```

The **show ip route** command displays information about the interfaces of each device, including their IP addresses and how they are connected to Device B:

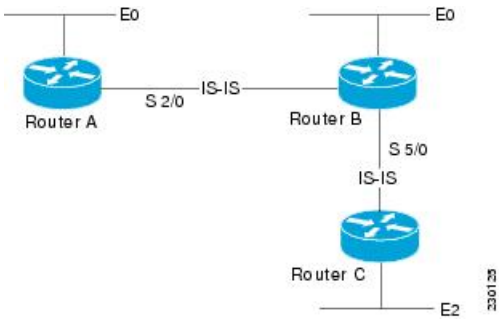
```
DeviceB# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     172.17.0.0/24 is subnetted, 1 subnets
C       172.17.1.0 is directly connected, Ethernet0/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Serial4/0
     172.21.0.0/24 is subnetted, 1 subnets
C       172.21.1.0 is directly connected, Serial5/0
     172.22.0.0/24 is subnetted, 1 subnets
i L1    172.22.1.0 [115/20] via 172.21.1.2, Serial5/0
     10.0.0.0/24 is subnetted, 1 subnets
i L1    10.1.1.0 [115/20] via 192.168.1.2, Serial2/0
C    192.168.1.0/24 is directly connected, Serial2/0
C    192.168.3.0/24 is directly connected, Serial3/0
```

The **show isis spf-log** command displays logs of Level 1 and Level 2 LSPs including time of occurrence, duration, count, and the event that triggered the LSP.

```
DeviceC## show isis spf-log

   level 1 SPF log
  When   Duration  Nodes  Count    First trigger LSP   Triggers
00:01:30       0      3      7        DeviceB.00-00  PERIODIC NEWADJ NEWLSP TLVT
   level 2 SPF log
  When   Duration  Nodes  Count    First trigger LSP   Triggers
00:01:31       0      3      7        DeviceB.00-00  PERIODIC NEWADJ NEWLSP TLVT
```

The figure below illustrates the sample configuration.

Figure 144: IS-IS Routing



Where to Go Next

- To initially configure and enable IS-IS, see the “Configuring a Basic IS-IS Network” module.
- To customize IS-IS for your network design, see the “Customizing IS-IS for Your Network Design” module.
- To customize IS-IS for achieving fast convergence and scalability, see the following modules:
  - “Overview of IS-IS Fast Convergence”
  - “Setting Best Practice Parameters for IS-IS Fast Convergence”
  - “Reducing Failure Detection Times in IS-IS Networks”
  - “Reducing Link Failure and Topology Change Notification Times in IS-IS Networks”
  - “Reducing Alternate-Path Calculation Times in IS-IS Networks”
- To enhance IS-IS network security, see the "Enhancing Security in an IS-IS Network" module.

Additional References for IS-IS Overview and Basic Configuration

Related Documents

Related Topic	Document Title
IPv6 Routing: IS-IS Support for IPv6	"IPv6 Routing: IS-IS Support for IPv6 " module
IPv6 Routing: Route Redistribution	"IPv6 Routing: Route Redistribution" module
IPv6 Routing: IS-IS Support for IPv6	"IPv6 Routing: IS-IS Support for IPv6 " module

Standards

Standard	Title
ISO 8473	*CLNP, Connectionless Network Protocol*
ISO 9542	*ES-IS Routing Information Exchange Protocol*
ISO/IEC 10589	*IS-IS Protocol*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1195	*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments* (http://www.ietf.org/rfc/rfc1195.txt)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IS-IS Overview and Basic Configuration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

area —A physically connected portion of a routing domain in which all devices are assigned a common area address. Also known as the Level-1 subdomain. A routing domain may consist of multiple areas that are reachable by traversing the Level-2 subdomain.

area address —The high-order octets of the Network Entity Title (NET) assigned to an IS. All ISs in the same Level-1 area are assigned the same area address.

CLNP —ISO Connectionless Network Protocol as defined in ISO 8473.

DIS —Designated Intermediate System. An IS elected by all the ISs operating on a multiaccess circuit at a given level to represent the multiaccess circuit. The DIS sends pseudonode LSPs on behalf of the circuit advertising adjacencies to all the ISs operating on that circuit.

domain —The portion of a network on which the IS-IS protocol is configured to operate. The routing domain consists of all Level-1 areas and the Level-2 subdomain.

ES —end system. An ES is any nonrouting host or node.

Integrated IS-IS —Extended form of IS-IS that supports multiple network protocols. Extensions have been defined in IETF documents, especially RFC 1195.

IS —intermediate system. OSI term for a device.

IP —Internet Protocol Version 4, also known as IPv4.

IPv6 —Internet Protocol Version 6.

IS-IS —Intermediate System-to-Intermediate System. Routing protocol as defined in ISO/IEC 10589.

Level-1 router —An IS that supports Level-1 routing for its assigned area.

Level-2 router —An IS that supports Level-2 routing.

Level-2 subdomain —All Level-2 capable devices in a domain and the links that interconnect them. Level-1 areas are interconnected via the Level-2 subdomain. For routing in a domain to work properly, the Level-2 subdomain must not be partitioned.

NET —Network Entity Title. An address assigned to an instance of the IS-IS protocol. The NET includes an area address, a system ID, and an N-selector. When multiple NETs are assigned to an IS-IS instance, only the area address portion of the NET may differ.

NSEL —N-selector. The least significant octet of a Network Entity Title. It is always assigned the value 00.

system ID —The part of the NET that immediately follows the area address. The field is 6 octets long.

CHAPTER 156

IPv6 Routing: Route Redistribution

IPv6 route redistribution supports redistributing routes into an IPv6 IS-IS routing process and redistributing IPv6 IS-IS routes between IS-IS levels.

Information About IPv6 Routing: Route Redistribution

IS-IS Enhancements for IPv6

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and OSI routes. Extensions to the IS-IS command-line interface (CLI) allow configuration of IPv6-specific parameters. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4.

IS-IS in IPv6 supports either single-topology mode or multiple topology mode.

IPv6 IS-IS Route Redistribution

IS-IS for IPv6 supports redistributing routes into an IPv6 IS-IS routing process and redistributing IPv6 IS-IS routes between IS-IS levels.

Preserving Metrics During Redistribution

When ISIS redistributes a route, the prefix can be preserved as the original route installed in the routing information base (RIB) by using the options **rib-metric-as-external** or **rib-metric-as-internal** for the **metric-type** keyword in the **redistribute** command. The options are allowed when ISIS redistributes routes from any routing process, including another ISIS process.

How to Configure IPv6 Routing: Route Redistribution

Redistributing Routes into an IPv6 IS-IS Routing Process

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast**]
5. **redistribute** *source-protocol* [*process-id*] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-tag*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Device(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**address-family ipv6** [**unicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • **unicast**—(Optional) Specifies the unicast IPv6 unicast address family. This is the default option.
Step 5	**redistribute** *source-protocol* [*process-id*] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-tag*] **Example:** `Device(config-router-af)# redistribute bgp 64500 metric 100 route-map isismap`	Redistributes routes from the specified protocol into the IS-IS process. • *source-protocol*—Can be one of the following: **bgp**, **connected**, **isis**, **rip** or **static**. • *process-id*—(Optional) Routing process name. • **metric** *metric-value*—Redistributes routes based on the metric value. • **metric-type** *type-value*—Specifies the link type, which can be the following: **external** to set an external ISIS

	Command or Action	Purpose
		metric type, **internal** to set an internal ISIS metric type, **rib-metric-as-external** to set metric type to external and use the RIB metric, and **rib-metric-as-internal** to set metric type to internal and use the RIB metric.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Redistributing IPv6 IS-IS Routes Between IS-IS Levels

Perform this task to redistribute IPv6 routes learned at one IS-IS level into a different level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast**]
5. **redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} **distribute-list** *list-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Device(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**address-family ipv6** [**unicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • **unicast**—(Optional) Specifies the unicast IPv6 unicast address family. This is the default option.

	Command or Action	Purpose
Step 5	**redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} **distribute-list** *list-name* **Example:** `Device(config-router-af)# redistribute isis level-1 into level-2`	Redistributes IPv6 routes from one IS-IS level into another IS-IS level. • By default, the routes learned by Level 1 instances are redistributed by the Level 2 instance. **Note** The *protocol* argument must be **isis** in this configuration of the **redistribute** command. Only the arguments and keywords relevant to this task are specified here.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Verifying IPv6 IS-IS Configuration and Operation

SUMMARY STEPS

1. **enable**
2. **show ipv6 protocols** [**summary**]
3. **show isis** [*process-tag*] [**ipv6** | *****] **topology**
4. **show clns** [*process-tag*] **neighbors** *interface-type interface-number*] [**area**] [**detail**]
5. **show clns** *area-tag* **is-neighbors** [*type number*] [**detail**]
6. **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
7. **show isis ipv6 rib** [*ipv6-prefix*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 protocols** [**summary**] **Example:** `Device# show ipv6 protocols`	Displays the parameters and current state of the active IPv6 routing processes.
Step 3	**show isis** [*process-tag*] [**ipv6** | *****] **topology** **Example:** `Device# show isis topology`	Displays a list of all connected routers running IS-IS in all areas.

	Command or Action	Purpose
Step 4	**show clns** [*process-tag*] **neighbors** *interface-type interface-number*] [**area**] [**detail**] **Example:** `Device# show clns neighbors detail`	Displays end system (ES), intermediate system (IS), and multitopology IS-IS (M-ISIS) neighbors.
Step 5	**show clns** *area-tag* **is-neighbors** [*type number*] [**detail**] **Example:** `Device# show clns is-neighbors detail`	Displays IS-IS adjacency information for IS-IS neighbors. • Use the **detail** keyword to display the IPv6 link-local addresses of the neighbors.
Step 6	**show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:** `Device# show isis database detail`	Displays the IS-IS link-state database. • In this example, the contents of each LSP are displayed using the **detail** keyword.
Step 7	**show isis ipv6 rib** [*ipv6-prefix*] **Example:** `Device# show isis ipv6 rib`	Displays the IPv6 local RIB.

Configuration Examples for IPv6 Routing: Route Redistribution

Example: Redistributing Routes into an IPv6 IS-IS Routing Process

The following example redistributes IPv6 BGP routes into the IPv6 IS-IS Level 2 routing process:

```
router isis
 address-family ipv6
 redistribute bgp 64500 metric 100 route-map isismap
 exit
```

Example: Redistributing IPv6 IS-IS Routes Between IS-IS Levels

The following example redistributes IPv6 IS-IS Level 1 routes into the IPv6 IS-IS Level 2 routing process:

```
router isis
 address-family ipv6
 redistribute isis level-1 into level-2
```

Example: Configuring IS-IS for IPv6

In the following example, output information about the parameters and current state of that active IPv6 routing processes is displayed using the **show ipv6 protocols**command:

```
Device# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "isis"
  Interfaces:
    GigabitEthernet0/0/3
    GigabitEthernet0/0/1
    Serial1/0/1
    Loopback1 (Passive)
    Loopback2 (Passive)
    Loopback3 (Passive)
    Loopback4 (Passive)
    Loopback5 (Passive)
  Redistribution:
    Redistributing protocol static at level 1
  Address Summarization:
    L2: 2001:DB8:33::/16  advertised with metric 0
    L2: 2001:DB8:44::/16  advertised with metric 20
    L2: 2001:DB8:66::/16  advertised with metric 10
    L2: 2001:DB8:77::/16  advertised with metric 10
```

In the following example, output information about all connected routers running IS-IS in all areas is displayed using the **show isis topology**command:

```
Device# show isis topology
IS-IS paths to level-1 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.000C
0000.0000.000D  20      0000.0000.00AA  Se1/0/1         *HDLC*
0000.0000.000F  10      0000.0000.000F  GE0/0/1         0050.e2e5.d01d
0000.0000.00AA  10      0000.0000.00AA  Se1/0/1         *HDLC*
IS-IS paths to level-2 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.000A  10      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000B  20      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000C  --
0000.0000.000D  30      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000E  30      0000.0000.000A  GE0/0/3         0010.f68d.f063
```

In the following example, output information to confirm that the local router has formed all the necessary IS-IS adjacencies with other IS-IS neighbors is displayed using the **show clns is-neighbors**command. To display the IPv6 link-local addresses of the neighbors, specify the **detail** keyword.

```
Device# show clns is-neighbors detail
System Id      Interface   State  Type Priority  Circuit Id         Format
0000.0000.00AA Se1/0/1     Up     L1   0         00                 Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::YYYY:D37C:C854:5
  Uptime: 17:21:38
0000.0000.000F Et0/0/1     Up     L1   64        0000.0000.000C.02  Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::XXXX:E2FF:FEE5:D01D
  Uptime: 17:21:41
0000.0000.000A Et0/0/3     Up     L2   64        0000.0000.000C.01  Phase V
```

```
  Area Address(es): 49.000b
  IPv6 Address(es): FE80::ZZZZ:F6FF:FE8D:F063
  Uptime: 17:22:06
```

In the following example, detailed output information that displays both end system (ES) and intermediate system (IS) neighbors is displayed using the **show clns neighbors** command with the **detail** keyword.

```
Device# show clns neighbors detail
System Id          Interface    SNPA            State  Holdtime  Type Protocol
0000.0000.0007     GE3/3        aa00.0400.6408  UP     26        L1   IS-IS
Area Address(es): 20
IP Address(es): 172.16.0.42*
Uptime: 00:21:49
0000.0C00.0C35     GE3/2        0000.0c00.0c36  Up     91        L1   IS-IS
Area Address(es): 20
IP Address(es): 192.168.0.42*
Uptime: 00:21:52
0800.2B16.24EA     GE3/3        aa00.0400.2d05  Up     27        L1   M-ISIS
Area Address(es): 20
IP Address(es): 192.168.0.42*
IPv6 Address(es): FE80::2B0:8EFF:FE31:EC57
Uptime: 00:00:27
0800.2B14.060E     GE3/2        aa00.0400.9205  Up     8         L1   IS-IS
Area Address(es): 20
IP Address(es): 192.168.0.30*
Uptime: 00:21:52
```

In the following example, detailed output information about LSPs received from other routers and the IPv6 prefixes they are advertising is displayed using the **show isis database**command with the **detail** keyword specified:

```
Device# show isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C00.0C35.00-00  0x0000000C   0x5696        325           0/0/0
  Area Address: 47.0004.004D.0001
  Area Address: 39.0001
  Metric: 10   IS 0000.0C00.62E6.03
  Metric: 0    ES 0000.0C00.0C35
 --More--
0000.0C00.40AF.00-00* 0x00000009   0x8452        608           1/0/0
  Area Address: 47.0004.004D.0001
  Topology: IPv4 (0x0) IPv6 (0x2)
  NLPID: 0xCC 0x8E
  IP Address: 172.16.21.49
  Metric: 10   IS 0800.2B16.24EA.01
  Metric: 10   IS 0000.0C00.62E6.03
  Metric: 0    ES 0000.0C00.40AF
  IPv6 Address: 2001:DB8::/32
  Metric: 10   IPv6 (MT-IPv6) 2001:DB8::/64
  Metric: 5    IS-Extended cisco.03
  Metric: 10   IS-Extended cisco1.03
  Metric: 10    IS (MT-IPv6) cisco.03
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
0000.0000.000A.00-00  0x00000059   0x378A        949               0/0/0
  Area Address: 49.000b
  NLPID:        0x8E
  IPv6 Address: 2001:DB8:1:1:1:1:1:1
  Metric: 10         IPv6 2001:DB8:2:YYYY::/64
  Metric: 10         IPv6 2001:DB8:3:YYYY::/64
```

```
  Metric: 10         IPv6 2001:DB8:2:YYYY::/64
  Metric: 10         IS-Extended 0000.0000.000A.01
  Metric: 10         IS-Extended 0000.0000.000B.00
  Metric: 10         IS-Extended 0000.0000.000C.01
  Metric: 0          IPv6 11:1:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:2:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:3:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:4:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:5:YYYY:1:1:1:1:1/128
0000.0000.000A.01-00  0x00000050   0xB0AF        491               0/0/0
  Metric: 0          IS-Extended 0000.0000.000A.00
  Metric: 0          IS-Extended 0000.0000.000B.00
```

The following example shows output from the **show isis ipv6 rib** command. An asterisk (*) indicates prefixes that have been installed in the primary IPv6 RIB as IS-IS routes. Following each prefix is a list of all paths in order of preference, with optimal paths listed first and suboptimal paths listed after optimal paths.

```
Device# show isis ipv6 rib

IS-IS IPv6 process "", local RIB
  2001:DB8:88:1::/64
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L2  metric 20 LSP [3/7]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L2  metric 20 LSP [3/7]
* 2001:DB8:1357:1::/64
    via FE80::202:7DFF:FE1A:9471/GigabitEthernet2/1/0, type L2  metric 10 LSP [4/9]
* 2001:DB8:45A::/64
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L1  metric 20 LSP [C/6]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L1  metric 20 LSP [C/6]
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L2  metric 20 LSP [3/7]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L2  metric 20 LSP [3/7]
```

Additional References for IPv6 Routing: Route Redistribution

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
IP Routing ISIS commands	Cisco IOS IP Routing: ISIS Command Reference
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
ISIS overview	*IS-IS Overview and Basic Configuration*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: Route Redistribution

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 157

IPv6 Routing: IS-IS Support for IPv6

This module describes how to configure Integrated Intermediate System-to-Intermediate System (IS-IS) for IPv6. IS-IS is an Interior Gateway Protocol (IGP) that advertises link-state information throughout the network to create a picture of the network topology. IS-IS is an Open Systems Interconnection (OSI) hierarchical routing protocol that designates an intermediate system as a Level 1 or Level 2 device. Level 2 devices route between Level 1 areas to create an intradomain routing backbone. Integrated IS-IS uses a single routing algorithm to support several network address families, such as IPv6, IPv4, and OSI.

Information About IPv6 Routing: IS-IS Support for IPv6

IS-IS Enhancements for IPv6

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and OSI routes. Extensions to the IS-IS command-line interface (CLI) allow configuration of IPv6-specific parameters. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4.

IS-IS in IPv6 supports either single-topology mode or multiple topology mode.

IS-IS Single-Topology Support for IPv6

Single-topology support for IPv6 allows IS-IS for IPv6 to be configured on interfaces along with other network protocols (for example, IPv4 and Connectionless Network Service [CLNS]). All interfaces must be configured with the identical set of network address families. In addition, all routers in the IS-IS area (for Level 1 routing) or the domain (for Level 2 routing) must support the identical set of network layer address families on all interfaces.

When single-topology support for IPv6 is being used, either old- or new-style TLVs may be used. However, the TLVs used to advertise reachability to IPv6 prefixes use extended metrics. Cisco routers do not allow an interface metric to be set to a value greater than 63 if the configuration is not set to support only new-style

TLVs for IPv4. In single-topology IPv6 mode, the configured metric is always the same for both IPv4 and IPv6.

IPv6 IS-IS Local RIB

A router that is running IS-IS IPv6 maintains a local RIB in which it stores all routes to destinations it has learned from its neighbors. At the end of each SPF, IS-IS attempts to install the best (that is, the least-cost) routes to a destination present in the local RIB in the global IPv6 routing table.

How to Configure IPv6 Routing: IS-IS Support for IPv6

Configuring Single-Topology IS-IS for IPv6

Configuring IS-IS comprises two activities. The first activity creates an IS-IS routing process and is performed using protocol-independent IS-IS commands. The second activity in configuring IPv6 IS-IS configures the operation of the IS-IS protocol on an interface.

Before you begin

Before configuring the router to run IPv6 IS-IS, globally enable IPv6 using the **ipv6 unicast-routing** global configuration command.

Note If you are using IS-IS single-topology support for IPv6, IPv4, or both IPv6 and IPv4, you may configure both IPv6 and IPv4 on an IS-IS interface for Level 1, Level 2, or both Level 1 and Level 2. However, if both IPv6 and IPv4 are configured on the same interface, they must be running the same IS-IS level. That is, IPv4 cannot be configured to run on IS-IS Level 1 only on a specified GigabitEthernet or FastEthernet interface while IPv6 is configured to run IS-IS Level 2 only on the same GigabitEthernet or FastEthernet interface.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **net** *network-entity-title*
5. **exit**
6. **interface** *type number*
7. **ipv6 address** {*ipv6-address / prefix-length | prefix-name sub-bits/prefix-length*
8. **ipv6 router isis** *area-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**net** *network-entity-title* **Example:** `Router(config-router)# net 49.0001.0000.0000.000c.00`	Configures an IS-IS network entity title (NET) for the routing process. • The *network-entity-title* argument defines the area addresses for the IS-IS area and the system ID of the router.
Step 5	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies the interface type and number, and enters interface configuration mode.
Step 7	**ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits/prefix-length* **Example:** `Router(config-if)# ipv6 address 2001:DB8::3/64`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface. **Note** Refer to the Implementing IPv6 Addressing and Basic Connectivity module for more information on configuring IPv6 addresses.
Step 8	**ipv6 router isis** *area-name* **Example:** `Router(config-if)# ipv6 router isis area2`	Enables the specified IPv6 IS-IS routing process on an interface.

Customizing IPv6 IS-IS

Perform this task to configure a new administrative distance for IPv6 IS-IS, configure the maximum number of equal-cost paths that IPv6 IS-IS will support, configure summary prefixes for IPv6 IS-IS, and configure an IS-IS instance to advertise the default IPv6 route (::/0). It also explains how to configure the hold-down

period between partial route calculations (PRCs) and how often Cisco IOS XE software performs the SPF calculation when using multitopology IS-IS.

You can customize IS-IS multitopology for IPv6 for your network, but you likely will not need to do so. The defaults for this feature are set to meet the requirements of most customers and features. If you change the defaults, refer to the IPv4 configuration guide and the IPv6 command reference to find the appropriate syntax.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **default-information originate** [**route-map** *map-name*]
6. **distance** *value*
7. **maximum-paths** *number-paths*
8. **summary-prefix** *ipv6-prefix prefix-length* **level-1** | **level-1-2**| **level-2**]
9. **prc-interval** *seconds* [*initial-wait*] [*secondary-wait*
10. **spf-interval** [**level-1** | **level-2**] *seconds initial-wait*] [*secondary-wait*
11. **exit**
12. **interface** *type number*
13. **isis ipv6 metric** *metric-value* [**level-1** | **level-2** | **level-1-2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**address-family ipv6** [**unicast**	**multicast**] **Example:** `Router(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the **unicast** keyword is not specified with the **address-family ipv6** command.

	Command or Action	Purpose
Step 5	**default-information originate** [**route-map** *map-name*] **Example:** `Router(config-router-af)# default-information originate`	(Optional) Injects a default IPv6 route into an IS-IS routing domain. • The **route-map** keyword and *map-name* argument specify the conditions under which the IPv6 default route is advertised. • If the **route map** keyword is omitted, then the IPv6 default route will be unconditionally advertised at Level 2.
Step 6	**distance** *value* **Example:** `Router(config-router-af)# distance 90`	(Optional) Defines an administrative distance for IPv6 IS-IS routes in the IPv6 routing table. • The *value*argument is an integer from 10 to 254. (The values 0 to 9 are reserved for internal use).
Step 7	**maximum-paths** *number-paths* **Example:** `Router(config-router-af)# maximum-paths 3`	(Optional) Defines the maximum number of equal-cost routes that IPv6 IS-IS can support. • This command also supports IPv6 Border Gateway Protocol (BGP) and Routing Information Protocol (RIP). • The *number-paths* argument is an integer from 1 to 64. The default for BGP is one path; the default for IS-IS and RIP is 16 paths.
Step 8	**summary-prefix** *ipv6-prefix prefix-length* **level-1** | **level-1-2**| **level-2**] **Example:** `Router(config-router-af)# summary-prefix 2001:DB8::/24`	(Optional) Allows a Level 1-2 router to summarize Level 1 prefixes at Level 2, instead of advertising the Level 1 prefixes directly when the router advertises the summary. • The *ipv6-prefix* argument in the **summary-prefix** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. • The *prefix-length* argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 9	**prc-interval** *seconds* [*initial-wait*] [*secondary-wait* **Example:** `Router(config-router-af)# prc-interval 20`	(Optional) Configures the hold-down period between PRCs for multitopology IS-IS for IPv6.
Step 10	**spf-interval** [**level-1** | **level-2**] *seconds initial-wait*] [*secondary-wait* **Example:**	(Optional) Configures how often Cisco IOS XE software performs the SPF calculation for multitopology IS-IS for IPv6.

	Command or Action	Purpose
	`Router(config-router-af)# spf-interval 30`	
Step 11	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode, and returns the router to router configuration mode. • Repeat this step to exit router configuration mode and return the router to global configuration mode.
Step 12	**interface** *type number* **Example:** `Router(config-router)# interface GigabitEthernet 0/0/1`	Specifies the interface type and number, and enters interface configuration mode.
Step 13	**isis ipv6 metric** *metric-value* [**level-1** | **level-2** | **level-1-2** **Example:** `Router(config-if)# isis ipv6 metric 20`	(Optional) Configures the value of an multitopology IS-IS for IPv6 metric.

Disabling IPv6 Protocol-Support Consistency Checks

Perform this task to disable protocol-support consistency checks in IPv6 single-topology mode.

For single-topology IS-IS IPv6, routers must be configured to run the same set of address families. IS-IS performs consistency checks on hello packets and will reject hello packets that do not have the same set of configured address families. For example, a router running IS-IS for both IPv4 and IPv6 will not form an adjacency with a router running IS-IS for IPv4 or IPv6 only. In order to allow adjacency to be formed in mismatched address-families network, the **adjacency-check** command in IPv6 address family configuration mode must be disabled.

Note Entering the **no adjacency-check** command can adversely affect your network configuration. Enter the **no adjacency-check** command only when you are running IPv4 IS-IS on all your routers and you want to add IPv6 IS-IS to your network but you need to maintain all your adjacencies during the transition. When the IPv6 IS-IS configuration is complete, remove the **no adjacency-check**command from the configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **no adjacency-check**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Device(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**address-family ipv6** [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 5	**no adjacency-check** **Example:** `Device(config-router-af)# no adjacency-check`	Disables the IPv6 protocol-support consistency checks performed on hello packets, allowing IPv6 to be introduced into an IPv4-only network without disrupting existing adjacencies. • The **adjacency-check**command is enabled by default.

Disabling IPv4 Subnet Consistency Checks

Perform this task to disable IPv4 subnet consistency checking when forming adjacencies. Cisco IOS XE software historically makes checks on hello packets to ensure that the IPv4 address is present and has a consistent subnet with the neighbor from which the hello packets are received. To disable this check, use the **no adjacency-check** command in the router configuration mode. However, if multitopology IS-IS is configured, this check is automatically suppressed, because multitopology IS-IS requires routers to form an adjacency regardless of whether or not all routers on a LAN support a common protocol.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **no adjacency-check**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Device(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**no adjacency-check** **Example:** `Device(config-router-af)# no adjacency-check`	Disables the IPv6 protocol-support consistency checks performed on hello packets, allowing IPv6 to be introduced into an IPv4-only network without disrupting existing adjacencies. • The **adjacency-check**command is enabled by default.

Verifying IPv6 IS-IS Configuration and Operation

SUMMARY STEPS

1. **enable**
2. **show ipv6 protocols** [**summary**]
3. **show isis** [*process-tag*] [**ipv6** | *****] **topology**
4. **show clns** [*process-tag*] **neighbors** *interface-type interface-number*] [**area**] [**detail**]
5. **show clns** *area-tag* **is-neighbors** [*type number*] [**detail**]
6. **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
7. **show isis ipv6 rib** [*ipv6-prefix*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 protocols** [**summary**] **Example:**	Displays the parameters and current state of the active IPv6 routing processes.

	Command or Action	Purpose
	`Device# show ipv6 protocols`	
Step 3	**show isis** [*process-tag*] [**ipv6** | *****] **topology** **Example:** `Device# show isis topology`	Displays a list of all connected routers running IS-IS in all areas.
Step 4	**show clns** [*process-tag*] **neighbors** *interface-type interface-number*] [**area**] [**detail**] **Example:** `Device# show clns neighbors detail`	Displays end system (ES), intermediate system (IS), and multitopology IS-IS (M-ISIS) neighbors.
Step 5	**show clns** *area-tag* **is-neighbors** [*type number*] [**detail**] **Example:** `Device# show clns is-neighbors detail`	Displays IS-IS adjacency information for IS-IS neighbors. • Use the **detail** keyword to display the IPv6 link-local addresses of the neighbors.
Step 6	**show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:** `Device# show isis database detail`	Displays the IS-IS link-state database. • In this example, the contents of each LSP are displayed using the **detail** keyword.
Step 7	**show isis ipv6 rib** [*ipv6-prefix*] **Example:** `Device# show isis ipv6 rib`	Displays the IPv6 local RIB.

Configuration Examples for IPv6 Routing: IS-IS Support for IPv6

Example: Customizing IPv6 IS-IS

The following example advertises the IPv6 default route (::/0)--with an origin of GigabitEthernet interface 0/0/1--with all other routes in router updates sent on GigabitEthernet interface 0/0/1. This example also sets an administrative distance for IPv6 IS-IS to 90, defines the maximum number of equal-cost paths that IPv6 IS-IS will support as 3, and configures a summary prefix of 2001:DB8::/24 for IPv6 IS-IS.

```
router isis
 address-family ipv6
 default-information originate
 distance 90
 maximum-paths 3
 summary-prefix 2001:DB8::/24
 exit
```

Example: Disabling IPv6 Protocol-Support Consistency Checks

The following example disables the **adjacency-check** command to allow a network administrator to configure IPv6 IS-IS on the router without disrupting the existing adjacencies:

```
router isis
 address-family ipv6
 no adjacency-check
```

Example: Configuring IS-IS for IPv6

In the following example, output information about the parameters and current state of that active IPv6 routing processes is displayed using the **show ipv6 protocols**command:

```
Device# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "isis"
  Interfaces:
    GigabitEthernet0/0/3
    GigabitEthernet0/0/1
    Serial1/0/1
    Loopback1 (Passive)
    Loopback2 (Passive)
    Loopback3 (Passive)
    Loopback4 (Passive)
    Loopback5 (Passive)
  Redistribution:
    Redistributing protocol static at level 1
  Address Summarization:
    L2: 2001:DB8:33::/16  advertised with metric 0
    L2: 2001:DB8:44::/16  advertised with metric 20
    L2: 2001:DB8:66::/16  advertised with metric 10
    L2: 2001:DB8:77::/16  advertised with metric 10
```

In the following example, output information about all connected routers running IS-IS in all areas is displayed using the **show isis topology**command:

```
Device# show isis topology
IS-IS paths to level-1 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.000C
0000.0000.000D  20      0000.0000.00AA  Se1/0/1         *HDLC*
0000.0000.000F  10      0000.0000.000F  GE0/0/1         0050.e2e5.d01d
0000.0000.00AA  10      0000.0000.00AA  Se1/0/1         *HDLC*
IS-IS paths to level-2 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.000A  10      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000B  20      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000C  --
0000.0000.000D  30      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000E  30      0000.0000.000A  GE0/0/3         0010.f68d.f063
```

In the following example, output information to confirm that the local router has formed all the necessary IS-IS adjacencies with other IS-IS neighbors is displayed using the **show clns is-neighbors**command. To display the IPv6 link-local addresses of the neighbors, specify the **detail** keyword.

```
Device# show clns is-neighbors detail
System Id      Interface   State  Type Priority  Circuit Id         Format
0000.0000.00AA Se1/0/1     Up     L1   0         00                 Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::YYYY:D37C:C854:5
  Uptime: 17:21:38
0000.0000.000F Et0/0/1     Up     L1   64        0000.0000.000C.02  Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::XXXX:E2FF:FEE5:D01D
  Uptime: 17:21:41
0000.0000.000A Et0/0/3     Up     L2   64        0000.0000.000C.01  Phase V
  Area Address(es): 49.000b
  IPv6 Address(es): FE80::ZZZZ:F6FF:FE8D:F063
  Uptime: 17:22:06
```

In the following example, detailed output information that displays both end system (ES) and intermediate system (IS) neighbors is displayed using the **show clns neighbors** command with the **detail** keyword.

```
Device# show clns neighbors detail
System Id          Interface    SNPA            State  Holdtime  Type Protocol
0000.0000.0007     GE3/3        aa00.0400.6408  UP     26        L1   IS-IS
Area Address(es): 20
IP Address(es): 172.16.0.42*
Uptime: 00:21:49
0000.0C00.0C35     GE3/2        0000.0c00.0c36  Up     91        L1   IS-IS
Area Address(es): 20
IP Address(es): 192.168.0.42*
Uptime: 00:21:52
0800.2B16.24EA     GE3/3        aa00.0400.2d05  Up     27        L1   M-ISIS
Area Address(es): 20
IP Address(es): 192.168.0.42*
IPv6 Address(es): FE80::2B0:8EFF:FE31:EC57
Uptime: 00:00:27
0800.2B14.060E     GE3/2        aa00.0400.9205  Up     8         L1   IS-IS
Area Address(es): 20
IP Address(es): 192.168.0.30*
Uptime: 00:21:52
```

In the following example, detailed output information about LSPs received from other routers and the IPv6 prefixes they are advertising is displayed using the **show isis database**command with the **detail** keyword specified:

```
Device# show isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C00.0C35.00-00  0x0000000C   0x5696        325           0/0/0
  Area Address: 47.0004.004D.0001
  Area Address: 39.0001
  Metric: 10   IS 0000.0C00.62E6.03
  Metric: 0    ES 0000.0C00.0C35
 --More--
0000.0C00.40AF.00-00* 0x00000009   0x8452        608           1/0/0
  Area Address: 47.0004.004D.0001
  Topology: IPv4 (0x0) IPv6 (0x2)
  NLPID: 0xCC 0x8E
  IP Address: 172.16.21.49
  Metric: 10   IS 0800.2B16.24EA.01
  Metric: 10   IS 0000.0C00.62E6.03
  Metric: 0    ES 0000.0C00.40AF
  IPv6 Address: 2001:DB8::/32
```

```
  Metric: 10   IPv6 (MT-IPv6) 2001:DB8::/64
  Metric: 5    IS-Extended cisco.03
  Metric: 10   IS-Extended cisco1.03
  Metric: 10    IS (MT-IPv6) cisco.03
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
0000.0000.000A.00-00  0x00000059   0x378A        949               0/0/0
  Area Address: 49.000b
  NLPID:        0x8E
  IPv6 Address: 2001:DB8:1:1:1:1:1:1
  Metric: 10         IPv6 2001:DB8:2:YYYY::/64
  Metric: 10         IPv6 2001:DB8:3:YYYY::/64
  Metric: 10         IPv6 2001:DB8:2:YYYY::/64
  Metric: 10         IS-Extended 0000.0000.000A.01
  Metric: 10         IS-Extended 0000.0000.000B.00
  Metric: 10         IS-Extended 0000.0000.000C.01
  Metric: 0          IPv6 11:1:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:2:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:3:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:4:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:5:YYYY:1:1:1:1:1/128
0000.0000.000A.01-00  0x00000050   0xB0AF        491               0/0/0
  Metric: 0          IS-Extended 0000.0000.000A.00
  Metric: 0          IS-Extended 0000.0000.000B.00
```

The following example shows output from the **show isis ipv6 rib** command. An asterisk (*) indicates prefixes that have been installed in the primary IPv6 RIB as IS-IS routes. Following each prefix is a list of all paths in order of preference, with optimal paths listed first and suboptimal paths listed after optimal paths.

```
Device# show isis ipv6 rib

IS-IS IPv6 process "", local RIB
  2001:DB8:88:1::/64
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L2  metric 20 LSP [3/7]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L2  metric 20 LSP [3/7]
* 2001:DB8:1357:1::/64
    via FE80::202:7DFF:FE1A:9471/GigabitEthernet2/1/0, type L2  metric 10 LSP [4/9]
* 2001:DB8:45A::/64
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L1  metric 20 LSP [C/6]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L1  metric 20 LSP [C/6]
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L2  metric 20 LSP [3/7]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L2  metric 20 LSP [3/7]
```

Additional References

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of Cisco IS-IS conceptual information with links to all the individual IS-IS modules	"Integrated IS-IS Routing Protocol Overview"

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: IS-IS Support for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 158

Configuring Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters

The Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters feature allows you to disable the Integrated Intermediate System-to-Intermediate System (IS-IS) protocol at the interface level or at the global IS-IS process level without removing the IS-IS configuration parameters.

This module describes the tasks to configure and monitor a basic Intermediate System-to-Intermediate System (IS-IS) network. The IS-IS process and adjacency formation are also explained. IS-IS is link-state protocol that allows the network designer to organize the network into a group of flooding domains. Often deployed as the Interior Gateway Protocol (IGP) for an ISP network backbone, IS-IS is capable of handling large topologies and large numbers of routing changes.

Prerequisites for Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters

- Before performing the tasks in this module, you should be familiar with the concepts described in the “Integrated IS-IS Routing Protocol Overview” module.
- You should know your network design and how you want traffic to flow through it before configuring IS-IS. Define areas, prepare an addressing plan for the devices (including defining the NETs), and determine the interfaces that will run Integrated IS-IS. To facilitate verification, a matrix of adjacencies should be prepared before you configure your devices, showing what neighbors should be expected in

the adjacencies table. For more information about verifying IS-IS configuration and formed adjacencies, see “Monitoring IS-IS”.

Information About Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters

IS-IS Process and Adjacencies

IS-IS requires some configuration on both the device and the interface. An IS-IS process is created when you enable IS-IS on a device and define a specific tag to identify that routing process. Interfaces configured with a specific tag will be part of the corresponding device process. More than one IS-IS process can run on a device for Connectionless Network Service (CLNS), but only one IS-IS process can run for IP.

Small IS-IS networks are built as a single area that includes all the devices in the network. As the network grows larger, it is usually reorganized into a backbone area made up of the connected set of all Level 2 devices from all areas. The areas are connected to local areas. Within a local area, devices know how to reach all system IDs. Between areas, devices know how to reach the backbone, and the backbone devices know how to reach other areas.

Devices establish Level 1 adjacencies to perform routing within a local area (intra-area routing). Devices establish Level 2 adjacencies to perform routing between Level 1 areas (inter-area routing).

If the network administrator does not specify Level 1 or Level 2 routing for the routing process being configured, the default routing behavior for the routing process will be Level 1-2.

If Level 2 routing is configured on any process, additional processes are automatically configured as Level 1, with the exception of previously configured Level 2 process, which will remain Level 2. You can have only one Level-2 process. You can configure the Level-2 process to perform Level-1 routing at the same time. If Level-2 routing is not desired for a device instance, use the **is-type** command in device configuration mode to remove the Level-2 capability. You can also use the **is-type** command to configure a different device instance as a Level-2 device.

Some networks use legacy equipment that supports only Level 1 routing. These devices are typically organized into many small areas that cannot be aggregated due to performance limitations. Cisco devices are used to interconnect each area to the Level 2 backbone.

Network entity titles (NETs) define the area addresses and the system ID of the device. See the “Configuring ISO CLNS” module in the *Cisco IOS ISO CLNS Configuration Guide* for a more detailed discussion of NETs.

PDU Packet Types in IS-IS Routing

The OSI stack defines a unit of data as a protocol data unit (PDU). A frame therefore is regarded by OSI as a data-link PDU, and a packet is regarded as a network PDU. There are four types of PDU packets, and each type can be Level 1 or Level 2:

- LSP—Link-state PDU. Used to distribute link-state information.
- IIH PDU—For IS-IS this is called the IS-IS Hello PDU. Used to establish and maintain adjacencies.

Note On point-to-point links, IIH PDUs will be the same for Level 1 and Level 2. Both Level-1 and Level-2 IIH use the same type of PDU, but they carry different circuit types.

- PSNP—Partial sequence numbers protocol data unit (PDU). Used to acknowledge and request link-state information.
- CSNP—Complete sequence number protocol data unit (PDU). Used to distribute the complete link-state database of a device.

IS-IS LSPs include specific information about the device's attachments. The following information is included in multiple Type Length Value (TLV) fields in the main body of the LSP:

- The links to neighbor device intermediate systems (ISs), including the metrics of those interfaces
- The links to the neighbor end systems (ESs)

How to Create, Monitor and Make Changes to Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters

Enabling IS-IS as an IP Routing Protocol on the Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **net** *network-entity-title*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Assigns a tag to an IS-IS process. Enters router configuration mode. • Configure tags to identify multiple IS-IS processes by giving a meaningful name for each routing process. If the tag is not specified, a null tag (0) is assumed and the process is referenced with a null tag. The tag name must be unique among all IP router processes for the device.
Step 4	**net** *network-entity-title* **Example:** `Device(config-router)# net 49.0001.0000.0000.000b.00`	Configures the NET on the device. • The NET identifies the device for IS-IS.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Enabling IS-IS as an IP Routing Protocol on the Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [*secondary*]
5. **ip router isis [***area-tag***]**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:**	Enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:** `Device(config-if)# ip address 172.16.1.27 255.255.255.0`	Sets the primary IP address on the interface.
Step 5	**ip router isis [***area-tag***]** **Example:** `Device(config-if)# ip router isis company1`	Enables IS-IS on the interfaces that are to use IS-IS to distribute their IP information (and additionally that might be used to establish IS-IS adjacencies). • Use the *area-tag* argument to specify to which IS-IS process the device belongs. • If there is more than one IS-IS process on the device, repeat the **ip router isis** command for each interface, specifying an area tag for each interface to associate each interface with the specific process to which it belongs.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring IS-IS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isis display delimiter** [**return** *count* | **character** *count*]
4. **exit**
5. **show ip protocols**
6. **show clns** *area-tag* **is-neighbors** [*type number*] [**detail**]
7. **show clns interface** [*type number*]
8. **show clns** *area-tag* **neighbors** [*type number*] [**area**] [**detail**]
9. **show clns** *area-tag* **traffic**
10. **show ip route** [*ip-address* [*mask*]] [[**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**]]
11. **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
12. **show isis database verbose**
13. **show isis lsp-log**
14. **show isis** [*area-tag*] [**ipv6** | *****] **spf-log**

15. **show isis** [*process-tag*] [**ipv6** | *****] **topology**
16. **show isis** [*area-tag*] **neighbors** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**isis display delimiter** [**return** *count* | **character** *count*] **Example:** `Device(config)# isis display delimiter return 3`	Makes output from multiarea displays easier to read by specifying the delimiter to use to separate displays of information.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip protocols** **Example:** `Device# show ip protocols`	Displays the parameters and current state of the active routing protocol process. • You can use this command to learn what protocols are active, what interfaces they are active on, what networks they are routing for, and other parameters that relate to the routing protocols.
Step 6	**show clns** *area-tag* **is-neighbors** [*type number*] [**detail**] **Example:** `Device# show clns is-neighbors detail`	Displays IS-IS information for IS-IS device adjacencies.
Step 7	**show clns interface** [*type number*] **Example:** `Device# show clns interface`	List the CLNS-specific information about each interface.
Step 8	**show clns** *area-tag* **neighbors** [*type number*] [**area**] [**detail**] **Example:** `Device# show clns area3 neighbors`	Displays both ES and IS neighbors. • The **show clns neighbor** command output verifies that the right adjacencies have established. A matrix of adjacencies should be prepared before you configure your devices, showing what neighbors

	Command or Action	Purpose
		should be expected in the adjacencies table, to facilitate verification.
Step 9	**show clns** *area-tag* **traffic** **Example:** `Device# show clns area3 traffic`	Displays traffic statistics. To monitor IS-IS for stability once it has been deployed across your network, enter the **show clns traffic** command to check the following important statistics: high numbers of SPFs, checksum errors, and retransmissions. To troubleshoot IS-IS behavior, you can use the output from the **show clns traffic** command to check for the following indicators: • The number of link-state PDUs (LSPs) can help you determine the stability of the IS-IS network. The number of LSPs should never be zero. However, an LSP count that keeps increasing over a short time period indicates a network issue. • LSP retransmissions should stay low. A later execution of the **show clns traffic** command that shows an increase in LSP retransmissions, as compared to an earlier execution of the command, can indicate instability or traffic problems. • To check for partial route calculations (PRCs), enter the **show clns traffic** command. PRCs are flooded when a change that does not affect topology is reported through an LSP; typical examples include the addition or removal of a prefix or metric changes for external or passive interfaces. A PRC update queue that remains full or increases to the maximum value for long periods of time indicates network instability. • LSP checksum errors indicate a problem. • The update queue should not stay full and should not drop much.
Step 10	**show ip route** [*ip-address* [*mask*]] [[**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**]] **Example:** `Device# show ip route 172.16.0.21`	Displays the current state of the routing table.
Step 11	**show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:**	Displays additional information about the IS-IS database. • Displays the link-state database for Level-1 and Level-2, the contents for each LSP, and the link-state protocol PDU identifier.

	Command or Action	Purpose
	`Device# show isis database detail`	
Step 12	**show isis database verbose** **Example:** `Device# show isis database verbose`	Displays additional information about the IS-IS database such as the sequence number, checksum, and holdtime for LSPs.
Step 13	**show isis lsp-log** **Example:** `Device# show isis lsp-log`	Displays a log of LSPs including time of occurrence, count, interface, and the event that triggered the LSP.
Step 14	**show isis** [*area-tag*] [**ipv6** | *****] **spf-log** **Example:** `Device# show isis spf-log`	Displays how often and why the device has run a full shortest path first (SPF) calculation. • If the device continues to run SPF without ceasing, there might be an issue regarding a change in the network (intra-area). The cause for the continued SPF calculations could be an interconnecting link that is transitioning up/down/up/down or a metric change. It is normal for the SPF calculation to run a few times when a network change occurs, but then it should cease.
Step 15	**show isis** [*process-tag*] [**ipv6** | *****] **topology** **Example:** `Device# show isis topology`	Displays a list of all connected devices in all areas.
Step 16	**show isis** [*area-tag*] **neighbors** [**detail**] **Example:** `Device# show isis neighbors detail`	Displays IS-IS adjacency information. • The **show isis neighbor detail** command output verifies that the right adjacencies have established. A matrix of adjacencies should be prepared before you configure your devices, showing what neighbors should be expected in the adjacencies table, to facilitate verification.

Example

When the **show isis neighbors** command is entered with the **detail** keyword, the output provides information about the IS-IS adjacencies that have formed.

```
Device1# show isis neighbors detail

System Id      Type Interface IP Address      State Holdtime Circuit Id
Device2         L2   Et1/0     10.1.1.0        UP   255      Circuit3.01
  Area Address(es): 32
  SNPA: aabb.cc00.2001
```

```
State Changed: 00:00:14
LAN Priority: 64
Format: Phase V
```

Troubleshooting Tips

You can use the following two system debugging commands to check your IS-IS IPv4 implementation.

- If adjacencies are not coming up properly, use the **debug isis adj-packets** command.
- To display a log of significant events during an IS-IS SPF calculation, use the **debug isis spf-events** command.

Shutting Down IS-IS to Make Changes to Your IS-IS Network

You can shut down IS-IS (placing it in an administrative down state) to make changes to the IS-IS protocol configuration, without losing your configuration parameters. You can shut down IS-IS at the interface level or at the global IS-IS process level. If the device was rebooted when the protocol was turned off, the protocol would be expected to come back up in the disabled state. When the protocol is set to the administrative down state, network administrators are allowed to administratively turn off the operation of the IS-IS protocol without losing the protocol configuration, to make a series of changes to the protocol configuration without having the operation of the protocol transition through intermediate—and perhaps undesirable—states, and to then reenable the protocol at a suitable time.

Before the introduction of the Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters feature, there was no nondestructive way to disable IS-IS operation. The only way to disable IS-IS at the device level was to issue the **no router isis** command, which removes the IS-IS configuration. At the interface level there are two ways to disable IS-IS operation. You can enter the **no ip router isis** command to remove IS-IS from the specified interface, or you can put the interface into passive mode such that the IP address of the specified interface will still be advertised. In either case, the current IS-IS configuration will be removed.

Shutting Down IS-IS in Interface Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **isis protocol shutdown**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Configures an interface and enters interface configuration mode.
Step 4	**isis protocol shutdown** **Example:** `Device(config-if)# isis protocol shutdown`	Disables the IS-IS protocol so that it cannot form adjacencies on a specified interface and places the IP address of the interface into the LSP that is generated by the device.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Shutting Down IS-IS in Router Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **protocol shutdown**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Device(config)# router isis 1`	Enables the IS-IS routing protocol and specifies an IS-IS process. • Enters router configuration mode.

	Command or Action	Purpose
Step 4	**protocol shutdown** **Example:** `Device(config-router)# protocol shutdown`	Prevents IS-IS from forming any adjacency on any interface and clears the IS-IS LSP database, without actually removing the IS-IS configuration.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuration Examples for Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters

Example: Configuring a Basic IS-IS Network

The following example shows how to configure three devices to run IS-IS as an IP routing protocol.

Device A Configuration

```
router isis
 net 49.0001.0000.0000.000a.00
interface ethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
interface serial 2/0
 ip router isis
 ip address 192.168.1.2 255.255.255.0
```

Device B Configuration

```
router isis
 net 49.0001.0000.0000.000b.00
interface ethernet0/0
 ip router isis
 ip address 172.17.1.1 255.255.255.0
interface serial2/0
 ip router isis
 ip address 192.168.1.1 255.255.255.0
interface serial5/0
 ip router isis
 ip address 172.21.1.1 255.255.255.0
```

Device C Configuration

```
router isis
 net 49.0001.0000.0000.000c.00
interface ethernet2/0
 ip router isis
```

```
 ip address 172.21.1.2 255.255.255.0
interface serial5/0
 ip router isis
 ip address 172.22.1.1 255.255.255.0
```

The **show isis topology** command displays the following information about how the devices are connected within the IS-IS network:

```
DeviceB# show isis topology

IS-IS paths to level-1 routers
System Id            Metric     Next-Hop             Interface   SNPA
DeviceA              10         DeviceA              Se2/0       *HDLC*
DeviceB              --
DeviceC              10         DeviceC              Se5/0       *HDLC*
IS-IS paths to level-2 routers
System Id            Metric     Next-Hop             Interface   SNPA
DeviceA              10         DeviceA              Se2/0       *HDLC*
DeviceB              --
DeviceC              10         DeviceC              Se5/0       *HDLC*
```

The **show isis database** command displays following information for the Level 1 and Level 2 LSPs for each device in the IS-IS network.

```
DeviceB# show isis database

IS-IS Level-1 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
DeviceA.00-00           0x00000005   0x1A1D        1063              0/0/0
DeviceB.00-00         * 0x00000006   0xD15B        1118              0/0/0
DeviceC.00-00           0x00000004   0x3196        1133              1/0/0
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
DeviceA.00-00           0x00000008   0x0BF4        1136              0/0/0
DeviceB.00-00         * 0x00000008   0x1701        1137              0/0/0
DeviceC.00-00           0x00000004   0x3624        1133              0/0/0
```

The **show ip route** command displays information about the interfaces of each device, including their IP addresses and how they are connected to Device B:

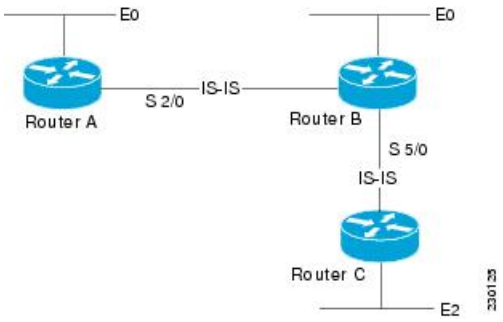
```
DeviceB# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     172.17.0.0/24 is subnetted, 1 subnets
C       172.17.1.0 is directly connected, Ethernet0/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Serial4/0
     172.21.0.0/24 is subnetted, 1 subnets
C       172.21.1.0 is directly connected, Serial5/0
     172.22.0.0/24 is subnetted, 1 subnets
i L1    172.22.1.0 [115/20] via 172.21.1.2, Serial5/0
     10.0.0.0/24 is subnetted, 1 subnets
i L1    10.1.1.0 [115/20] via 192.168.1.2, Serial2/0
C    192.168.1.0/24 is directly connected, Serial2/0
C    192.168.3.0/24 is directly connected, Serial3/0
```

The **show isis spf-log** command displays logs of Level 1 and Level 2 LSPs including time of occurrence, duration, count, and the event that triggered the LSP.

```
DeviceC## show isis spf-log

   level 1 SPF log
  When   Duration  Nodes  Count    First trigger LSP   Triggers
00:01:30       0      3      7        DeviceB.00-00  PERIODIC NEWADJ NEWLSP TLVT
   level 2 SPF log
  When   Duration  Nodes  Count    First trigger LSP   Triggers
00:01:31       0      3      7        DeviceB.00-00  PERIODIC NEWADJ NEWLSP TLVT
```

The figure below illustrates the sample configuration.

Figure 145: IS-IS Routing



Example: Shutting Down IS-IS in Interface Mode

The following device output shows that the device has two IS-IS adjacencies:

```
Device# show clns neighbors

System Id  Interface  SNPA            State  Holdtime  Type      Protocol
first      Et3/1      0002.7dd6.1c21  Up     25        L1L2      IS-IS
second     Et3/2      0004.6d25.c056  Up     29        L1L2      IS-IS
```

When the **isis protocol shutdown** command is entered for Ethernet interface 3/1, the IS-IS protocol will be disabled for the specified interface:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#
Device(config-if)# isis protocol shutdown
Device(config-if)# end
```

The following device output shows that the adjacency for Ethernet interface 3/1 has not formed:

```
Device# show clns neighbors

System Id  Interface  SNPA            State  Holdtime  Type      Protocol
second     Et3/2      0004.6d25.c056  Up     27        L1L2      IS-IS
```

Example: Shutting Down IS-IS in Router Mode

The following device output shows that the device has two IS-IS adjacencies:

```
Device# show clns neighbors

System Id  Interface  SNPA            State  Holdtime  Type      Protocol
south      Et3/1      0002.7dd6.1c21  Up     29        L1L2      IS-IS
north      Et3/2      0004.6d25.c056  Up     28        L1L2      IS-IS
```

The **protocol shutdown** command is entered so that IS-IS is disabled and no adjacencies will be formed on any interface:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# router isis area1
Device(config-router)# protocol shutdown
Device(config-router)# end
```

The following device output now shows that both adjacencies are gone.

```
Device# show clns neighbors

System Id  Interface  SNPA            State  Holdtime  Type      Protocol
```

When the **no protocol shutdown** command is entered, the adjacencies will again be formed on both interfaces:

```
Device(config)# router isis area1
Device(config-router)# no protocol shutdown
Device(config-router)# end
Device# show clns neighbors

System Id  Interface  SNPA            State  Holdtime  Type      Protocol
south      Et3/1      0002.7dd6.1c21  Up     24        L1L2      IS-IS
north      Et3/2      0004.6d25.c056  Up     24        L1L2      IS-IS
```

""Where to Go Next

- To customize IS-IS for your network design, see the "Customizing IS-IS for Your Network Design" module.
- To customize IS-IS for achieving fast convergence and scalability, see the following modules:
  - “Overview of IS-IS Fast Convergence”
  - “Setting Best Practice Parameters for IS-IS Fast Convergence”
  - “Reducing Failure Detection Times in IS-IS Networks”
  - “Reducing Link Failure and Topology Change Notification Times in IS-IS Networks”
  - “Reducing Alternate-Path Calculation Times in IS-IS Networks”
- To enhance IS-IS network security, see the “Enhancing Security in an IS-IS Network” module.

Additional References for Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of IS-IS concepts	"Integrated IS-IS Routing Protocol Overview" module
Customizing IS-IS for achieving fast convergence and scalability	"Overview of IS-IS Fast Convergence" module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
• CISCO-IETF-IP-FORWARD-MIB • CISCO-IETF-IP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1195	Use of OSI IS-IS for Routing in TCP/IP and Dual Environments *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments* (http://www.ietf.org/rfc/rfc1195.txt)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Integrated IS-IS Protocol Shutdown Support Maintaining Configuration Parameters

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 159

Customizing IS-IS for Your Network Design

This module describes optional tasks that you can perform to customize Intermediate System-to-Intermediate System (IS-IS) for your network design. You can optimize network traffic flow by setting metrics, specifying an IS-IS system type, summarizing addresses, generating a default route, and configuring a global default metric.

Prerequisites for Customizing IS-IS for Your Network Design

- Before performing the tasks in this module, you should be familiar with the concepts described in the "Integrated IS-IS Routing Protocol Overview" module.
- You should understand the concept of IP addressing. For more information on IP addressing, see the "Configuring IPv4 Addresses" chapter of the *Cisco IOS XE IP Addressing Services Configuration Guide*, Release 2.
- You should know your network design and how you want traffic to flow through it before configuring IS-IS. Define areas, prepare an addressing plan for the routers (including defining the network entity titles [NETs]), and determine the interfaces that will run Integrated IS-IS.
- IS-IS must be enabled.

Information About Customizing IS-IS for Your Network Design

You can enhance network traffic flow by configuring IS-IS metric values for Level-1 or Level-2 routing, in order to prioritize traffic through certain paths. You can customize network traffic flow by changing the metric cost for a specified interface. All IS-IS links use the metric of 10 by default. The protocol does not automatically incorporate link attributes such as bandwidth or delay when metric values are assigned. The total cost to a destination is the sum of the costs on all outgoing interfaces along a particular path from the source to the destination. The least-cost paths are preferred.

On multi-access networks, IS-IS elects a router to act as a pseudo-node representing the multi-access circuit. The elected router is known as the designated intermediate system (DIS). The DIS issues pseudo-node LSPs listing all of the routers which are reachable on the network. Each router on the network advertises in its non-pseudonode LSPs reachability to the DIS. This reduces the amount of information that needs to be advertised. A DIS is elected for each level that is operating on the network, for example both Level 1 and Level 2. By default, all routers have the same priority for being elected DIS. The MAC address of each router's interface onto the network is used as the tiebreaker. When all routers have the same priority, the addition or removal of a router onto the network can result in a chance in the DIS. This churn can be prevented by assigning a higher priority to the router which you wish to act as the DIS. Priorities can be configured individually for Level 1 and Level 2. By default the priority is 64. You can configure the priority in the range from 0 to 127.

You can configure a summary address to represent summarized (aggregate) addresses within the IS-IS routing table. This process is called route summarization. Using a summary address can enhance scalability and network stability because it reduces the amount of information that needs to be advertised and reduces the frequency of updates required. For example, a single route flap may not cause the summary advertisement to flap. The disadvantage of using the summary addresses is that routing may be sub-optimal, for example, the path to a specific destination covered by the summary address may be longer than it would have been, had all the individual addresses been advertised. Summary addresses are most commonly used to summarize routes from one Level-one area into the Level-2 subdomain. One summary address can include multiple groups of addresses for a given level. Routes learned from other routing protocols can also be summarized. The metric used to advertise the summary is the smallest metric of all the more-specific routes.

Whenever you specifically configure redistribution of routes into an IS-IS routing domain, the Cisco IOS XE software does not, by default, redistribute the default route into the IS-IS routing domain. If you wish to advertise a default route you must use the **default-information originate** command. This command causes a default route to be advertised by the router. Advertisement of the default route can be made conditional by using a route map. You can use the route map to identify the level into which the default route is to be announced, whether a particular non-default prefix must be reachable, etc.

In Cisco IOS XE software, IS-IS has a default metric value of 10 for all active interfaces. If the interface is passive, the default value is zero. Rather than change the metric values for the active interfaces one by one, you can configure a different default metric value to be used by all interfaces. All interfaces that had the original IS-IS default metric 10 will be configured with the new default value. Besides offering the user the convenience of being able to globally configure the value for all IS-IS interfaces, the feature helps prevent errors that may occur when interfaces are individually configured to change the metric value. For example the user may remove configured metrics from an interface, thereby restoring the default metric value of 10--perhaps unintentionally making that interface a highly preferred one in the network. Such an occurrence on the wrong interface could mean the rerouting of traffic across the network on an undesirable path.

Note The MTU size (1418 bytes) in GRE tunnel is less than default ISIS LSP-MTU of 1492 bytes. To configure ISIS IPv6 over GRE tunnels, you must update the ISIS LSP-MTU size in all routers when overlay MTU size is less than 1492 bytes in GRE tunnel.

Enhancing Your IS-IS Network Design at the Interface Level

Setting the IS-IS Link-State Metrics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type name*
4. **isis metric** *default-metric* [**level-1** | **level-2**]
5. **end**
6. **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type name* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Enters interface configuration mode.
Step 4	**isis metric** *default-metric* [**level-1**	**level-2**] **Example:** `Router(config-if)# isis metric 15 level-1`	Configures the metric for an interface. **Note** We highly recommend that you configure the metrics on all interfaces. If you do not do so, all links will have the same cost and the cost to reach any node in the network will be logically equivalent to the number of hops.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:**	(Optional) Displays the IS-IS link-state database. • To display information about each LSP and the link-state database, enter the **detail** keyword.

	Command or Action	Purpose
	`Router# show isis database detail`	

Prioritizing Designated Intermediate Systems for IS-IS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type name*
4. **isis priority** *number-value* [**level-1** | **level-2**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type name* **Example:** `Router(config)# interface gigabitethernet 0/3/0`	Enters interface configuration mode.
Step 4	**isis priority** *number-value* [**level-1**	**level-2**] **Example:** `Router(config-if)# isis priority 2 level-1`	Configures the priority used in the designated router election.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enhancing Your IS-IS Network Design at the Router Level

Limiting Level 1 and Level 2 Operations on the IS-IS Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **is-type** {**level-1** | **level-1-2** | **level-2-only**}
5. **end**
6. **show isis** [**ipv6**] [*] **topology**[**level-1**] [**level-2**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis 1`	Enables IS-IS as an IP routing protocol. • Enters router configuration mode.
Step 4	**is-type** {**level-1**	**level-1-2**	**level-2-only**} **Example:** `Router(config-router)# is-type level-1`	Configures the routing level for an instance of the IS-IS routing process. **Note** By default, Cisco IOS XE software enables both Level 1 and Level 2 operations on IS-IS routers. To specify that a router is to operate only as an area router (Level 1) or only as a backbone router (Level 2), use the **is-type** command. Specifying routers to act as Level 1, Level 2, or Level 1 and 2 can streamline your network design.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show isis** [**ipv6**] [*] **topology**[**level-1**] [**level-2**] **Example:** `Router# show isis topology level-1`	(Optional) Displays a list of all connected routers in all areas. • To confirm paths to all Level 1 or Level 2 routers in the area or areas in which this router resides, enter the **level-1** or **level-2** keywords, respectively.

Examples

The following example shows output from the **show isis topology** command for a router within a dual CLNS-IP network. In this example, because neither the **level-1** nor **level-2** optional keywords were entered, information is displayed for both Level 1 and Level 2 routers.

```
Router# show isis topology
Tag L2BB:
IS-IS paths to level-2 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.0005  --
0000.0000.0009  10      0000.0000.0009  Tu529           *Tunnel*
0000.0000.0017  20      0000.0000.0009  Tu529           *Tunnel*
0000.0000.0053  30      0000.0000.0009  Tu529           *Tunnel*
0000.0000.0068  20      0000.0000.0009  Tu529           *Tunnel*
Tag A3253-01:
IS-IS paths to level-1 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.0003  10      0000.0000.0003  FE1/0/0         0000.0c03.6944
0000.0000.0005  --
0000.0000.0053  10      0000.0000.0053  FE1 /0/0        0060.3e58.ccdb
```

Summarizing Address Ranges in the IS-IS Routing Table

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **summary-address** *address mask* {**level-1** | **level-1-2** | **level-2**}[**tag** *tag-number*] [**metric** *metric-value*]
5. **end**
6. **show isis database verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis 1`	Enables IS-IS as an IP routing protocol. • Enters router configuration mode.
Step 4	**summary-address** *address mask* {**level-1** | **level-1-2** | **level-2**} [**tag** *tag-number*] [**metric** *metric-value*] **Example:** `Router(config-router)# summary-address 10.1.0.0 255.255.0.0 level-2`	Creates aggregate addresses for IS-IS. **Note** Multiple groups of addresses can be summarized for a given level. Routes learned from other routing protocols can also be summarized. The metric used to advertise the summary is the smallest metric of all the more-specific routes. This command helps reduce the size of the routing table.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 6	**show isis database verbose** **Example:** `Router# show isis database verbose`	(Optional) Displays detailed information about the IS-IS database.

Generating an IS-IS Default Route

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **default-information originate** [**route-map** *map-name*]
5. **end**
6. **show ip route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis 1`	Enables IS-IS as an IP routing protocol. • Enters router configuration mode.
Step 4	**default-information originate** [**route-map** *map-name*] **Example:** `Router(config-router)# default-information originate`	Generates a default route into an IS-IS routing domain.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 6	**show ip route** **Example:** `Router# show ip route`	(Optional) Displays the current state of the routing table.

Configuring an IS-IS Default Metric

Note If you have already configured a metric for a specific interface by entering the **isis metric** command, the metric that has been configured for that specific interface will take precedence over any default set by the **metric** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **metric** *default-value* [**level-1** | **level-2**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis 1`	Enables IS-IS as an IP routing protocol. • Enters router configuration mode.
Step 4	**metric** *default-value* [**level-1** | **level-2**] **Example:** `Router(config-router)# metric 25 level-2`	Globally sets a new default metric value for all IS-IS interfaces. • The value 25 shown in the example will apply only to Level 2 IS-IS interfaces. If you do not enter the **level-1** or **level-2** keyword, the metric will be applied to both Level 1 and Level 2 IS-IS interfaces.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuration Examples for Customizing IS-IS for Your Network Design

Example Configuring a Global Default Metric for IPv4

The following configuration example for an IS-IS routing process called area1 sets a global default metric of 111 for the IS-IS interfaces:

```
interface gigabitethernet3/1/0
 ip address 172.16.10.2 255.255.0.0
 ip router isis area1
 no ip route-cache
 duplex half
!
interface gigabitethernet3/2/0
 ip address 192.168.242.2 255.255.255.0
```

```
 ip router isis area1
 no ip route-cache
 duplex half
router isis area1
 net 01.0000.0309.1234.00
 metric-style wide
 metric 111
```

In the following example, the **show clns interface** command confirms that the IS-IS IPv4 interface metric for both Level 1 and Level 2 interfaces is assigned the new default metric value 111:

```
Router# show clns interface
GigabitEthernet3/1/0 is up, line protocol is up
  Checksums enabled, MTU 1497, Encapsulation SAP
  ERPDUs enabled, min. interval 10 msec.
  CLNS fast switching enabled
  CLNS SSE switching disabled
  DEC compatibility mode OFF for this interface
  Next ESH/ISH in 39 seconds
  Routing Protocol: IS-IS
    Circuit Type: level-1-2
    Interface number 0x0, local circuit ID 0x1
    Level-1 Metric: 111, Priority: 64, Circuit ID: mekong.01
    Level-1 IPv6 Metric: 10
    Number of active level-1 adjacencies: 0
    Level-2 Metric: 111, Priority: 64, Circuit ID: mekong.01
    Level-2 IPv6 Metric: 10
    Number of active level-2 adjacencies: 0
    Next IS-IS LAN Level-1 Hello in 922 milliseconds
    Next IS-IS LAN Level-2 Hello in 1 seconds
GigabitEthernet3/2/0 is up, line protocol is up
  Checksums enabled, MTU 1497, Encapsulation SAP
  ERPDUs enabled, min. interval 10 msec.
  CLNS fast switching enabled
  CLNS SSE switching disabled
  DEC compatibility mode OFF for this interface
  Next ESH/ISH in 20 seconds
  Routing Protocol: IS-IS
    Circuit Type: level-1-2
    Interface number 0x1, local circuit ID 0x2
    Level-1 Metric: 111, Priority: 64, Circuit ID: mekong.02
    Level-1 IPv6 Metric: 10
    Number of active level-1 adjacencies: 1
    Level-2 Metric: 111, Priority: 64, Circuit ID: mekong.02
    Level-2 IPv6 Metric: 10
    Number of active level-2 adjacencies: 1
    Next IS-IS LAN Level-1 Hello in 2 seconds
    Next IS-IS LAN Level-2 Hello in 1 seconds
```

In the following example, the **isis metric** command is entered so that it will assign a metric value of 10. The metric value that is set with the **isis metric** command for GigabitEthernet interface 3/1/0 will take precedence over the metric value that was previously set with the **metric** command.

```
interface GigabitEthernet3/1/0
 ip address 172.30.10.2 255.255.0.0
 ip router isis area1
 no ip route-cache
 duplex half
 isis metric 10
!
interface GigabitEthernet3/2/0
 ip address 192.168.224.2 255.255.255.0
```

```
 ip router isis area1
 no ip route-cache
 duplex half
router isis area1
 net 01.0000.0309.1234.00
 metric-style wide
 metric 111
```

When the **show clns interface**command is entered, the router output confirms that the interface has an assigned IS-IS IPv4 metric value of 10:

```
Router# show clns interface
GigabitEthernet3/1/0 is up, line protocol is up
  Checksums enabled, MTU 1497, Encapsulation SAP
  ERPDUs enabled, min. interval 10 msec.
  CLNS fast switching enabled
  CLNS SSE switching disabled
  DEC compatibility mode OFF for this interface
  Next ESH/ISH in 53 seconds
  Routing Protocol: IS-IS
    Circuit Type: level-1-2
    Interface number 0x0, local circuit ID 0x1
    Level-1 Metric: 10, Priority: 64, Circuit ID: mekong.01
    Level-1 IPv6 Metric: 10
    Number of active level-1 adjacencies: 0
    Level-2 Metric: 10, Priority: 64, Circuit ID: mekong.01
    Level-2 IPv6 Metric: 10
    Number of active level-2 adjacencies: 0
    Next IS-IS LAN Level-1 Hello in 4 seconds
    Next IS-IS LAN Level-2 Hello in 4 seconds
GigabitEthernet3/2/0 is up, line protocol is up
  Checksums enabled, MTU 1497, Encapsulation SAP
  ERPDUs enabled, min. interval 10 msec.
  CLNS fast switching enabled
  CLNS SSE switching disabled
  DEC compatibility mode OFF for this interface
  Next ESH/ISH in 30 seconds
  Routing Protocol: IS-IS
    Circuit Type: level-1-2
    Interface number 0x1, local circuit ID 0x2
    Level-1 Metric: 111, Priority: 64, Circuit ID: mekong.02
    Level-1 IPv6 Metric: 10
    Number of active level-1 adjacencies: 1
    Level-2 Metric: 111, Priority: 64, Circuit ID: mekong.02
    Level-2 IPv6 Metric: 10
    Number of active level-2 adjacencies: 1
    Next IS-IS LAN Level-1 Hello in 2 seconds
    Next IS-IS LAN Level-2 Hello in 922 milliseconds
```

Additional References

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*

Related Topic	Document Title
Overview of Integrated IS-IS conceptual information with links to all the individual IS-IS modules	"Integrated IS-IS Routing Protocol Overview" module
Configuring IPv6	"Implementing IPv6 Addressing and Basic Connectivity" chapter in the *Cisco IOS IPv6 XE Configuration Guide*, Release 2
Configuring the IS-IS protocol for IPv6 networks	"Implementing IS-IS for IPv6" module in the *Cisco IOS XE IPv6 Configuration Guide* , Release 2
Customizing IS-IS for fast convergence and scalability	"Overview of IS-IS Fast Convergence" module
Enhancing IS-IS network security	"Enhancing Security in an IS-IS Network" module
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1195	*s*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Customizing IS-IS for Your Network Design

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 160

Segment Routing—IS-IS v4 node SID

The Segment Routing—ISIS v4 node SID feature provides support for segment routing on Cisco Intermediate System-to-Intermediate System (IS-IS) networks.

Information About Segment Routing IS-IS v4 Node SID

Segment Routing IS-IS v4 Node SID

Segment Routing relies on a small number of extensions to Cisco Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) protocols. There are two levels of configuration required to enable segment routing for a routing protocol instance. The top level segment routing configuration which is managed by segment routing infrastructure component enables segment routing, whereas, segment routing configuration at the router level enables segment routing for a specific address-family of a routing protocol instance. There are three segment routing states:

- SR_NOT_CONFIGURED
- SR_DISABLED
- SR_ENABLED

Segment routing configuration under the IGPs is allowed only if the SR state is either SR_DISABLED or SR_ENABLED. The SR_ENABLED state indicates that there is at least a valid SRGB range reserved through the MFI successfully. You can enable segment routing for IGPs under the router configuration sub mode, through commands. However, IGP segment routing are enabled only after the global SR is configured.

The SR_ENABLED is a necessary state for any protocol to enable SR, however, it is not a sufficient for enabling SR for a protocol instance. The reason being that the IS-IS still does not have any information about segment routing global block (SRGB) information. When the request to receive information about the SRGB is processed successfully, the IS-IS SR operational state is enabled.

Segment Routing requires each router to advertise its segment routing data-plane capability and the range of MPLS label values that are used for segment routing in the case where global SIDs are allocated. Data-plane

capabilities and label ranges are advertised using the SR-capabilities sub-TLV inserted into the IS-IS Router Capability TLV-242 that is defined in RFC4971.

ISIS SR-capabilities sub TLV includes all reserved SRGB ranges. However, the Cisco implementation supports only one SRGB range. The supported IPv4 prefix-SID sub TLV are TLV-135 and TLV-235.

How to Configure Segment Routing —IS-IS v4 Node SID

Configuring Segment Routing

Before you begin

Before configuring IS-IS to support segment routing you must first configure the segment routing feature in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **segment-routing mpls**
4. **connected-prefix-sid-map**
5. **address-family ipv4**
6. **10.1.1.1/32 index 100 range 1**
7. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**segment-routing mpls** **Example:** `Device(config-sr)# segment-routing mpls`	Enables the segment feature using the MPLS data plane.
Step 4	**connected-prefix-sid-map** **Example:** `Device(config-srmpls)# connected-prefix-sid-map`	Enters a sub-mode where you can configure address-family specific mappings for local prefixes and SIDs.

	Command or Action	Purpose
Step 5	**address-family ipv4** **Example:** `Device(config-srmpls-conn)# address-family ipv4`	Specifies IPv4 address prefixes.
Step 6	**10.1.1.1/32 index 100 range 1** **Example:** `Device(config-srmpls-conn-af)# 10.1.1.1/32 100 range 1`	Associates SID 100 with the address 10.1.1.1/32.
Step 7	**exit-address-family** **Example:** `Device(config-srmpls-conn-af)# exit-address-family`	Exits the address family.

Configuring Segment Routing on an IS-IS Network

Before you begin

Before you configure segment routing on IS-IS network, IS-IS must be enabled on your network.

SUMMARY STEPS

1. **router isis**
2. **net network-entity-title**
3. **metric-style wide**
4. **segment-routing** mpls
5. **exit**
6. **show isis segment-routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router isis** **Example:** `Device(config-router)# router isis`	Enables the IS-IS routing protocol and enters router configuration mode.
Step 2	**net network-entity-title** **Example:** `Device(config-router)# net 49.0000.0000.0003.00`	Configures network entity titles (NETs) for the routing instance.
Step 3	**metric-style wide**	Configures the device to generate and accept only wide link

	Command or Action	Purpose
	Example: `Device(config-router)# metric-style wide`	metrics.
Step 4	**segment-routing** mpls **Example:** `Device(config-router)# segment-routing mpls`	Configures segment routing operation state.
Step 5	**exit** **Example:** `Device(config-router)# exit`	Exits segment routing mode and returns to the configuration terminal mode.
Step 6	**show isis segment-routing** **Example:** `Device# show is-is segment-routing`	Displays the current state of the IS-IS segment routing.

Example

The following example displays output from the **show isis segment-routing state** command for the segment routing under IS-IS:

```
Device# show isis segment-routing

ISIS protocol is registered with MFI
ISIS MFI Client ID:0x63
Tag 1 - Segment-Routing:
   SR State:SR_ENABLED
   Number of SRGB:1
   SRGB Start:16000, Range:8000, srgb_handle:0x4500AED0, srgb_state: created
   Address-family IPv4 unicast SR is configured
     Operational state:Enabled
```

Configuring Prefix-SID for IS-IS

This section explains how to configure prefix segment identifier (SID) index under each interface.

Before you begin

Segment routing must be enabled on the corresponding address family.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **segment-routing mpls**
4. **connected-prefix-sid-map**

5. **address-family ipv4**
6. **10.1.1.1/32 index 100 range 1**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**segment-routing mpls** **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	**connected-prefix-sid-map** **Example:** `Device(config-srmpls)# connected-prefix-sid-map`	Enters a sub-mode where you can configure address-family specific mappings for local prefixes and SIDs.
Step 5	**address-family ipv4** **Example:** `Device(config-srmpls-conn)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 6	**10.1.1.1/32 index 100 range 1** **Example:** `Device(config-srmpls-conn-af)# 10.1.1.1/32 100 range 1`	Associates SID 100 with the address 10.1.1.1/32.
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits segment routing mode and returns to the configuration terminal mode.

Configuring Prefix Attribute N-Flag

By default, a flag called N-flag is set by IS-IS when advertising an SID that is associated with a loopback address. To clear this flag add explicit configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback3**
4. **isis prefix n-flag-clear**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback3** **Example:** `Device(config)# interface loopback3`	Specifies the interface loopback.
Step 4	**isis prefix n-flag-clear** **Example:** `Device(config-if)# isis prefix n-flag-clear`	Clears the prefix N-flag.

Configuring the Explicit Null Attribute

To disable penultimate-hop-popping (PHP) and add explicit-Null label, explicit-null option needs to be specified. Once the option is given, IS-IS sets the E flag in the prefix-SID sub TLV.

By default, a flag called E-flag (Explicit-Null flag) is set to 0 by ISIS when advertising a Prefix SID which is associated with a loopback address. If you wish to set this flag add explicit configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **segment-routing mpls**
4. **set-attributes**
5. **address-family ipv4**
6. **explicit-null**
7. **exit-address-family**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**segment-routing mpls** **Example:** `Device(config)# segment-routing mpls`	Configures segment routing mpls mode.
Step 4	**set-attributes** **Example:** `Device(config-srmpls)# set-attributes`	Sets the attribute.
Step 5	**address-family ipv4** **Example:** `Device(config-srmpls-attr)# address-family ipv4`	Specifies the IPv4 address family and enters router address family configuration mode.
Step 6	**explicit-null** **Example:** `Device(config-srmpls-attr-af)# explicit-null`	Enables the explicit-null label.
Step 7	**exit-address-family** **Example:** `Device(config-srmpls-attr-af)# exit-address-family`	Exits the address family.

Configuration Examples for Segment Routing —IS-IS v4 Node SID

Example: Configuring Segment Routing on IS-IS Network

The following example shows how to configure prefix segment identifier (SID) index under each interface:

```
Device(config)#segment-routing mpls
 Device(config-srmpls)#connected-prefix-sid-map
  Device(config-srmpls-conn)#address-family ipv4
   Device(config-srmpls-conn-af)#10.1.2.2/32 index 2 range 1
  Device(config-srmpls-conn-af)#exit-address-family
 Device(config-srmpls-conn-af)#end
```

Example: Configuring an Explicit Null Attribute

The following is an example of configuring an explicit null attribute:

```
Device(config)# segment-routing mpls
Device(config-srmpls)# set-attributes
 Device(config-srmpls-attr)# address-family ipv4
  Device(config-srmpls-attr-af)# explicit-null
 Device (config-srmpls-attr-af)# exit-address-family
```

Additional References for Segment Routing-IS-IS v4 Node SID

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html
IP Routing ISIS commands	Cisco IOS IP Routing ISIS commands http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Segment Routing with IS-IS v4 Node SID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

CHAPTER 161

IS-IS MIB

This feature introduces MIB support for the Intermediate System-to-Intermediate System (IS-IS) link-state routing protocol. IS-IS is used as the link-state routing protocol of choice by major service providers. The IS-IS MIB feature offers service providers an improved capability to continuously monitor the changing state of an IS-IS network by use of MIB objects to gather information relating to protocol parameters and trap notification objects that can signal the occurrence of significant protocol events such as an authentication failure or a mismatch in area addresses between Intermediate Systems (ISs). The protocol information collected by the IS-IS MIB objects and trap objects can be used by the network manager to derive statistics that can help monitor and improve overall network performance.

Prerequisites for IS-IS MIB

- Simple Network Management Protocol (SNMP) must be enabled on the router before notifications (traps) can be configured or before SNMP GET operations can be performed.
- IS-IS must be configured on the router.

Restrictions for IS-IS MIB

- All enhancements that are introduced by this feature are provided only by the Cisco private MIB CISCO-IETF-ISIS-MIB.my.
- The SNMP SET capability will not be supported for any IS-IS MIB objects. Objects with read-create or read-write access are understood to operate only as read-only.
- This feature is not supported for multiple instances of IS-IS.

Information About IS-IS MIB

Cisco IS-IS MIB Table Object Definitions

The IS-IS MIB feature introduces network management support for the IS-IS routing protocol through the use of IS-IS MIB table entries, MIB objects and MIB trap notification objects that comprise the Cisco private MIB CISCO-IETF-ISIS-MIB.my. New CLI has been added to enable SNMP notifications for IS-IS MIB objects. Notifications are provided for errors and other significant event information for the IS-IS network.

For more information on how to configure IS-IS MIB to receive the SNMP notifications, refer to the How to Enable IS-IS MIB, on page 2091.

The ciiManAreaAddrEntry table contains the set of area addresses manually configured for the IS. The ciiManAreaAddrEntry table defines the following MIB objects:

- ciiManAreaAddr
- ciiManAreaAddrExistState

The ciiAreaAddrEntry table groups sets of relevant area addresses reported in all Level 1 link-state packets (LSPs) that were generated or received by an IS from other ISs that are reachable through Level 1 routing.

Each entry contains one area address per LSP. The ciiAreaAddrEntry table defines the following MIB object:

- ciiAreaAddr

The ciiSysProtSuppEntry table contains a manually configured set of protocols supported by the IS. The supported protocol types are IPv4, IPv6, and ISO8473. The ciiSysProtSuppEntry table defines the following MIB objects:

- ciiSysProtSuppProtocol
- ciiSysProtSuppExistState

The ciiSummAddrEntry table contains a set of manually configured summary addresses used to form summarized IP TLVs originated by an ISS. This table is useful to combine and modify IP reachability announcements, and also controls leaking of L1 routes into L2. The ciiSummAddrEntry table defines the following MIB objects:

- ciiSummAddressType
- ciiSummAddress
- ciiSummAddrPrefixLen
- ciiSummAddrExistState
- ciiSummAddrMetric
- ciiSummAddrFullMetric

The ciiRedistributeAddrEntry table provides the criteria to decide if a route should be leaked from L2 to L1. When Domain Wide Prefix leaking is enabled (represented by ciiSysL2toL1Leaking), addresses that match the summary mask in the table are announced at L1 by routers. The Cisco MIB implementation also allows

retrieval of routes for masked entries based on configured access lists or route maps. The ciiRedistributeAddrEntry table defines the following MIB objects:

- ciiRedistributeAddrType
- ciiRedistributeAddrAddress
- ciiRedistributeAddrPrefixLen
- ciiRedistributeAddrExistState

The ciiRouterEntry table has one entry for every peer and it tracks the hostnames and Router IDs associated with that peer. The ciiRouterEntry table defines the following MIB objects.

- ciiRouterSysID
- ciiRouterLevel
- ciiRouterHostName
- ciiRouterID

Note The IS-IS MIB defines the ciiRouterLevel object to be the level of the IS. The Cisco implementation interprets the ciiRouterLevel object to be the level of the link-state packet (LSP) in which the hostname (ciiRouterHostName) and router ID (ciiRouterID) were received.

The ciiSysLevelEntry table captures level-specific information about the IS. This information includes parameters that control how LSPs are generated, metrics for SPF computation and the decision of whether to perform traffic engineering at this level.

The ciiSysLevelEntry table defines the following MIB objects:

- ciiSysLevelIndex
- ciiSysLevelOrigLSPBuffSize
- ciiSysLevelMinLSPGenInt
- ciiSysLevelOverloadState
- ciiSysLevelSetOverload
- ciiSysLevelSetOverloadUntil
- ciiSysLevelMetricStyle
- ciiSysLevelSPFConsiders
- ciiSysLevelTEEnabled

Note For the ciiSysLevelOverloadState MIB object, the Cisco MIB follows the correct interpretation of IS state transition per the future IETF draft MIB revisions. The draft-ietf-isis-wg-16.txt did not follow the ISO 10589:2002 definition correctly. Per the ISO 10589:2002 definition, the waiting state is defined for low memory resource condition and the overloaded state is enabled by the administrator. Moreover, the Cisco implementation does not support a transition to a waiting state on low memory.

The ciiCircEntry table contains circuit-specific information about each broadcast or point-to-point interface used in this IS-IS. Each entry is associated with a corresponding interface, based on the circuit type (broadcast or point-to-point interfaces). In other words, only interfaces that are configured as broadcast or point-to-point can be polled. The Cisco implementation of the IS-IS MIB does not support the following circuit types: staticIn, staticOut, dA (dynamically assigned). The ciiCircEntry table defines the following MIB objects:

- ciiCircIndex
- ciiCircIfIndex
- ciiCircIfSubIndex
- ciiCircAdminState
- ciiCircExistState
- ciiCircType
- ciiCircExtDomain
- ciiCircLevel
- ciiCircPassiveCircuit
- ciiCircMeshGroupEnabled
- ciiCircMeshGroup
- ciiCircSmallHellos
- ciiCircLastUpTime
- ciiCirc3WayEnabled
- ciiCircExtendedCircID

Note The ciiCircExtDomain MIB table object is not implemented because externalDomain linkage is not supported by Cisco IOS software.

The ciiNextCircIndex object, which is defined outside ciiCircTable, is used to assign a unique index value to the ciiCircIndex through a SET operation. The Cisco MIB implementation does not implement this object because the SET ability currently is not supported, and ciiCircIndex is determined uniquely through data from configured interfaces.

The ciiCircLevelEntry table contains level-specific information about IS-IS circuits. The ciiCircLevelEntry table contains the following MIB objects:

- ciiCircLevelIndex
- ciiCircLevelMetric
- ciiCircLevelWideMetric
- ciiCircLevelISPriority
- ciiCircLevelIDOctet
- ciiCircLevelID
- ciiCircLevelDesIS
- ciiCircLevelHelloMultiplier
- ciiCircLevelHelloTimer
- ciiCircLevelDRHelloTimer
- ciiCircLevelLSPThrottle
- ciiCircLevelMinLSPRetransInt
- ciiCircLevelCSNPInterval
- ciiCircLevelPartSNPInterval

The ciiSystemCounterEntry table has a sequence of entries used to track system-wide events using counters. The ciiSystemCounterEntry table defines the following MIB objects:

- ciiSysStatLevel
- ciiSysStatCorrLSPs
- ciiSysStatAuthTypeFails
- ciiSysStatAuthFails
- ciiSysStatLSPDbaseOloads
- ciiSysStatManAddrDropFromAreas
- ciiSysStatAttmptToExMaxSeqNums
- ciiSysStatSeqNumSkips
- ciiSysStatOwnLSPPurges
- ciiSysStatIDFieldLenMismatches
- ciiSysStatPartChanges
- ciiSysStatSPFRuns
- ciiSysStatLSPErrors

Note The ciiSysStatPartChanges object is not implemented because the ability to detect partition changes currently is not supported by Cisco IOS software.

The ciiCircuitCounterEntry table is used to track system-wide events specific to a circuit and level. The ciiCircuitCounterEntry table defines the following MIB objects:

- ciiCircuitType
- ciiCircAdjChanges
- ciiCircNumAdj
- ciiCircInitFails
- ciiCircRejAdjs
- ciiCircIDFieldLenMismatches
- ciiCircMaxAreaAddrMismatches
- ciiCircAuthTypeFails
- ciiCircAuthFails
- ciiCircLANDesISChanges

Note The ciiCircInitFails MIB object does not return any data because circuit initialization failures are not tracked by Cisco IOS software.

The ciiPacketCounterEntry table tracks the number of IS-IS packets sent and received over a circuit at one level. At any time, the traffic flow along one direction is recorded. All objects defined in this table are Counter objects. The ciiPacketCounterEntry table defines the following MIB objects:

- ciiPacketCountLevel
- ciiPacketCountDirection
- ciiPacketCountIIHellos
- ciiPacketCountISHellos
- ciiPacketCountESHellos
- ciiPacketCountLSPs
- ciiPacketCountCSNPs
- ciiPacketCountPSNPs
- ciiPacketCountUnknowns

Note The ciiPacketCountISHellos MIB object tracks the number of end system-Intermediate system (ES-IS) hellos only at system granularity and not at per-level or per-circuit.

- The ciiPacketCountESHellos MIB objects tracks the number of end-system (ES) hellos only at system granularity and not at per-level or per-circuit.

- The ciiPacketCountUnknowns MIB object can track only unknown packet types that are received, not those that are sent in any given level.

The ciiISAdjEntry table has one entry associated with every adjacency to an IS (in other words, a table of adjacencies).

However, this object cannot be used to track multiple adjacencies in a LAN, with each adjacency corresponding to a level. Thus the best priority level is selected among the configured objects.

The ciiISAdjEntry table defines the following MIB objects:

- ciiISAdjChanges
- ciiISAdjIndex
- ciiISAdjState
- ciiISAdj3WayState
- ciiISAdjNeighSNPAAddress
- ciiISAdjNeighSysType
- ciiISAdjNeighSysID
- ciiISAdjNbrExtendedCircID
- ciiISAdjUsage
- ciiISAdjHoldTimer
- ciiIsAdjNeighPriority
- ciiISAdjLastUpTime

Note The ciiISAdjChanges MIB object gathers information based on the best priority level that is selected among the configured objects, per the restriction against the software support of multiple adjacencies in a LAN for the ciiISAdjEntry table.

- The ciiISAdjNeighPriority MIB object gathers information based on the best priority level that is selected among the configured objects, per the restriction against the software support of multiple adjacencies in a LAN for the ciiISAdjEntry table.

The ciiISAdjAreaAddrEntry table contains entries for the sets of area addresses of neighboring ISs as reported in received IS-IS Hello protocol data units (PDU)s. The ciiISAdjAreaAddrEntry table defines the following MIB objects:

- ciiISAdjAreaAddrIndex
- ciiISAdjAreaAddress

The ciiISAdjIPAddrEntry table contains entries that are formed by a set of IP addresses of neighboring ISs as reported in received Hello PDUs. The ciiISAdjIPAddrEntry table defines the following MIB objects:

- ciiISAdjIPAddrIndex

- ciiISAdjIPAddrType
- ciiISAdjIPAddrAddress

The ciiISAdjProtSuppEntry table contains information about the protocols supported by neighboring ISs as reported in received Hello PDUs. The ciiISAdjProtSuppEntry table defines the following MIB object:

- ciiISAdjProtSuppProtocol

The ciiRAEntry table records information about a reachable NSAP or address prefix that is manually configured or learned dynamically.

The ciiRAEntry table defines the following MIB objects:

- ciiRAIndex
- ciiRAExistState
- ciiRAAdminState
- ciiRAAddrPrefix
- ciiRAMapType
- ciiRAMetric
- ciiRAMetricType
- ciiRASNPAAddress
- ciiRASNPAMask
- ciiRASNPAPrefix
- ciiRAType

Note The ciiRAMapType MIB Object supports only implicit (null) and explicit mapping types. The extractIDI and extractDSP types are not supported.

- Because the ciiRAMapType MIB Object does not support the extractIDI and extractDSP mapping types, the ciiraSNPAPrefix and ciiRASNPAMask MIB objects will hold no data, as they depend on the unsupported mapping types. The ciiRAMapType and ciiRASNPAMask MIB objects are not implemented.
- The ciiRAType MIB object does not support the manual creation of IP reachability addresses.

Each entry in the ciiIPRAEntry table records information about one IP reachable address manually configured on the IS or learned from another protocol. The ciiIPRAEntry table defines the following MIB objects:

- ciiIPRADestType
- ciiIPRADest
- ciiIPRADestPrefixLen
- ciiIPRANextHopIndex
- ciiIPRANextHopType

- ciiIPRANextHop
- ciiIPRAType
- ciiIPRAExistState
- ciiIPRAAdminState
- ciiIPRAMetric
- ciiIPRAMetricType
- ciiIPRAFullMetric
- ciiIPRASNPAAddress
- ciiIPRASourceType

Note The ciiIpRAType MIB object does not support manually created IP reachability addresses.

The ciiLSPSummaryEntry table (LSP Summary Table) provides LSP summary information.

The ciiLSPSummaryEntry table defines the following MIB objects:

- ciiLSPLevel
- ciiLSPID
- ciiLSPSeq
- ciiLSPZeroLife
- ciiLSPChecksum
- ciiLSPLifetimeRemain
- ciiLSPPDULength
- ciiLSPAttributes

The ciiLSPTLVEntry table provides a complete record of all LSPs as a sequence of {Type, Length, Value} tuples. The ciiLSPTLVEntry table defines the following MIB objects:

- ciiLSPTLVIndex
- ciiLSPTLVSeq
- ciiLSPTLVChecksum
- ciiLSPTLVType
- ciiLSPTLVLen
- ciiLSPTLVValue

Fields that are required for notifications are recorded in the ciiNotificationEntry table. The ciiNotificationEntry table is not meant for query since the MAX-ACCESS clause of the MIB objects is "accessible-for-notify." The information for notifications will be directly provided at the time of event generation. The following MIB

objects are used only in trap notifications where their value is determined and directly based on input parameters for the IS-IS trap generation process.

- ciiPduLspId
- ciiPduFragment
- ciiPduFieldLen
- ciiPduMaxAreaAddress
- ciiPduProtocolVersion
- ciiPduLspSize
- ciiPduOriginatingBufferSize
- ciiPduProtocolsSupported
- ciiAdjState
- ciiErrorOffset
- ciiErrorTLVType
- ciiNotifManualAddress
- ciiNotifIsLevelIndex

Note The MIB objects ciiNotifManualAddress and ciiNotifIsLevelIndex were added separately and are not defined in draft-ietf-isis-wg-mib-16.txt. These have been provided as a replacement for ciiManAreaAddr and ciiSysLevelIndex respectively to be used only in trap notifications. They have a MAX-ACCESS clause of "accessible-for-notify."

Cisco IS-IS MIB Trap Notifications

IS-IS MIB for Generic System-Wide Errors

The following MIB trap objects are for generic, system-wide errors that can occur in the IS-IS network:

- ciiManualAddressDrops--The ciiManualAddressDrops trap is generated when one of the manually configured area addresses assigned to the system is ignored while computing routes.
- ciiAuthenticationFailure--The ciiAuthenticationFailure trap is generated when the authenticating type information field in the PDU received from a circuit is incorrect. This is an edge-triggered notification.
- ciiIDLenMismatch--When an LSP with a different value of SystemID length is received, the ciiIDLenMismatch notification is generated specific to the circuit where the LSP was detected. This is an edge-triggered notification and hence will be generated only once for PDUs received on the same circuit.
- ciiMaxAreaAddressesMismatch--When the value of Maximum Area Addresses is changed in the LSP that is received from a circuit, the ciiMaxAreaAddressesMismatch trap notification is generated. The header of the packet is used to identify the cause of the mismatch in Maximum Area Address. This trap

is an edge-triggered notification and hence will be generated only once for PDUs received on the same circuit.

IS-IS MIB for LSP-Specific Errors

The following MIB trap objects are for LSP-specific errors that can occur in the IS-IS network:

- ciiCorruptedLSPDetected--When an LSP stored in memory is corrupted, the ciiCorruptedLSPDetected trap is generated.
- ciiAttemptToExceedMaxSequence--The ciiAttemptToExceedMaxSequence trap is generated each time a sequence number on a generated LSP wraps around the 32-bit sequence counter, forcing it to be purged and hence waiting for its reannouncement.
- ciiOwnLSPPurge--The ciiOwnLSPPurge trap is generated when a LSP is received from a circuit with your systemID and zero age.
- ciiSequenceNumberSkip--When an LSP is received without a SystemID or differing contents, the ciiSequenceNumberSkip trap is generated in order to increment the sequence number by 1.
- ciiAuthenticationTypeFailure--When an LSP is received from a circuit filled with a wrong authentication type field, the ciiAuthenticationTypeFailure notification is generated. This is an edge-triggered notification.
- ciiLSPTooLargeToPropagate--When an attempt is made to send an LSP over the circuit with a size greater than dataLinkBlockSize (link-specific parameter for maximum size of a data packet), the ciiLSPTooLargeToPropagate trap is generated indicating that the LSP could not be propagated. This is an edge-triggered notification and will be generated only once for all PDUs received on the same circuit.

Note Cisco IOS software does not support the condition that leads to this event. Therefore, this trap will not be generated.

- ciiOrigLSPBuffSizeMismatch--When an L1 or L2 LSP that has been received from a circuit has a size larger than the local value of ciiOriginatingBufferSize, or when an LSP has been received with the ciiOriginatingBufferSize option and there is a mismatch between local ciiOriginatingBufferSize and value of the PDU option field, this notification is generated. This is an edge-triggered notification and will be generated only once.

Note The originating buffer size TLV that is used to advertise this condition is not currently supported in Cisco IOS software and sufficient information to determine which condition caused the trap is not available. Therefore, this trap will not be generated.

- ciiProtocolsSupportedMismatch--The ciiProtocolsSupportedMismatch trap is generated when a non-pseudonode segment 0 LSP is received that does not have any matching protocols supported. This is an edge-triggered notification.

Note Cisco IOS software does not provide checks in the IS-IS implementation for detecting matching protocols in the case of received PDUs. The generation of the ciiProtocolsSupportedMismatch trap does not indicate a mismatch in protocols supported as specified in the protocol field of the received PDU.

- ciiLSPErrorDetected--The ciiLSPErrorDetected trap is generated to indicate that an LSP with a parse error has been received.

MIB Support for IS-IS Hello PDU-Specific Errors

The following MIB trap objects are for Hello PDU-specific errors that can occur in the IS-IS network:

- ciiVersionSkew--The ciiVersionSkew trap notification is generated when a Hello PDU is received from an IS running a different version of the IS-IS protocol. This is an edge-triggered notification and will be generated once for all PDUs received on the same circuit.
- ciiAreaMismatch--When a Hello PDU is received from an IS that does not share any area address, the ciiAreaMismatch notification is generated. This is an edge-triggered notification and will be generated only once for all PDUs received on the same circuit.
- ciiRejectedAdjacency--When a correct Hello PDU is received from an IS but adjacency is not established, the ciiRejectedAdjacency notification is generated to indicate that adjacency formation was not allowed. This is an edge-triggered notification.

MIB Support for IS-IS Transition State Changes

The following MIB trap objects are used to notify the network manager when a transition state change has occurred for an IS:

- ciiDatabaseOverload--The ciiDatabaseOverload trap object is used to notify the network manager when the system enters or leaves the Overload state.
- ciiAdjacencyChange--When an IS-IS adjacency changes its state to UP or moves out of this state, it causes the ciiAdjacencyChange trap notification to be generated.

You can enable SNMP notifications to be sent when IS-IS errors and mismatches related to invalid field values in PDUs are detected. Errors can be classified as generic (applied to all PDUs), LPS-related, and IS-IS Hello PDU-related. When you enter the **snmp-server enable traps isis errors** command without specifying any of the optional keywords and arguments, all IS-IS traps are enabled. You can enter specific keywords and arguments to enable certain traps. For more information on how to enable specific traps or groups of traps, refer to the **snmp-server enable traps isis**command page.

You can enable IS-IS traps for the following system-wide errors that apply to all PDUs:

- Authentication
- Authentication type
- System ID field length mismatch
- Manually-configured address drop
- Mismatch in maximum area address values

You can enable IS-IS traps for the following errors that apply specifically to IS-IS Hello PDUs:

- Adjacency creation failure
- Mismatch in the area addresses between ISs
- IS-IS protocol version mismatch

You can enable IS-IS traps for the following errors that apply specifically to LSPs:

- Mismatch in LSP and originating buffer size
- Attempt made to exceed a maximum sequence number
- LSP in-memory corruption with an invalid checksum
- Packet parse failure on a receiving circuit
- Protocol-supported mismatch for non-pseudonode LSP
- Invalid attempt to purge a the LSP of a local IS
- Propagation failure caused by an oversized LSP
- A system ID has been configured with a sequence number skip.

How to Enable IS-IS MIB

Configuring the Router to Send SNMP Notifications for IS-IS to a Host

Before you begin

SNMP must be enabled on your network.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth**| **priv**]}] *community-string* [**upd-port** *port*] [*notification-type*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. • If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth**| **priv**]}] *community-string* [**upd-port** *port*] [*notification-type*] **Example:** `Router(config)# snmp-server host 172.16.1.1 traps version 3 mycommunitystring isis`	Specifies the recipient (target host) for IS-IS SNMP notification operations. • If no *notification-type* is specified, all enabled notifications (traps or informs) will be sent to a specified host. If you want to send only IS-IS notifications to the specified host, you can use the optional **isis** keyword as the value for the *notification-type* argument. (See the example.)
Step 5	**end** **Example:** `Router(config)# end`	Ends your configuration sessions and exits global configuration mode.

Examples

The following example configures the router to send SNMP notifications for IS-IS to a host:

```
Router> enable
Router# configure terminal
Router(config)# snmp-server host 172.31.1.1 traps version 3 mycommunity string isis
```

What to Do Next

If you want to globally enable all IS-IS traps, refer to the Enabling All IS-IS Traps, on page 2092. If you want to enable groups of IS-IS traps, refer to the and the Enabling IS-IS State-Change Traps, on page 2095.

Enabling All IS-IS Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps isis**

4. **no snmp-server enable traps isis** [**errors** [*error-type*]] [**state-change** [*state-change-type*]]
5. **exit**
6. **show running-config** [*options*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps isis** **Example:** `Router(config)# snmp-server enable traps isis`	Enables all SNMP notifications defined in the IS-IS MIB. **Note** This step is required only if you wish to enable all IS-IS traps. To enable specific groups of traps, see the Enabling IS-IS Error Traps, on page 2094 or the Enabling IS-IS State-Change Traps, on page 2095. When you enter the **no snmp-server enable traps isis** command, all IS-IS traps will be disabled.
Step 4	**no snmp-server enable traps isis** [**errors** [*error-type*]] [**state-change** [*state-change-type*]] **Example:** `Router(config)# no snmp-server enable traps isis state-change database-overload`	Disables the sending of SNMP notifications for IS-IS state changes. **Note** This step is required only if you wish to disable a particular trap or set of traps. To enable specific groups of traps, see Enabling IS-IS Error Traps, on page 2094 or the Enabling IS-IS State-Change Traps, on page 2095.
Step 5	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 6	**show running-config** [*options*] **Example:** `Router# show running-config | include traps`	Displays the running configuration to verify which traps have been enabled.

Examples

The following example shows how to globally enable all IS-IS traps:

```
Router> enable
Router# configure terminal
Router(config)# snmp-server enable traps isis
```

What to Do Next

If you do not wish to enable all IS-IS traps, refer to the Enabling IS-IS Error Traps, on page 2094 for enabling one or more IS-IS error traps, or refer to the Enabling IS-IS State-Change Traps, on page 2095 for enabling one or more IS-IS state-change traps.

Enabling IS-IS Error Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps isis** [**errors** [*error-type*]] [**state-change** [*state-change-type*]]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps isis** [**errors** [*error-type*]] [**state-change** [*state-change-type*]] **Example:** `Router(config)# snmp-server enable traps isis errors lsp`	Enables SNMP notifications for IS-IS errors. • When you enter the **lsp** keyword for the *error-type*, only the LSP error traps are enabled. (See the **snmp-server enable traps isis** command in the *Cisco IOS IP Routing: ISIS Command Reference* for a list of *error-type* keywords.)
Step 4	**end** **Example:** `Router(config)# end`	Ends your configuration sessions and exits global configuration mode.

Examples

The following example shows how to enable only the IS-IS traps related to authentication errors:

```
Router> enable
Router# configure terminal
Router(config)# snmp-server enable traps isis errors authentication
```

Enabling IS-IS State-Change Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps isis** [**state-change** [*state-change-type*]]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps isis** [**state-change** [*state-change-type*]] **Example:** `Router(config)# snmp-server enable traps isis state-change`	Enables SNMP notifications for IS-IS state changes. **Note** When the **snmp-server enable traps isis state-change**command is entered without any of the optional keywords, both IS-IS state change traps are enabled. Entering the **no snmp-server enable traps isis state-change**command will disable both IS-IS state-change traps.
Step 4	**end** **Example:** `Router(config)# end`	Ends your configuration sessions and exits global configuration mode.

Examples

The following example shows how to enable only the IS-IS traps related to adjacency transition state changes:

```
Router> enable
Router# configure terminal
Router(config)# snmp-server enable traps isis state-change adjacency
```

Verifying IS-IS MIB Traps on the Router

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** [*options*] **Example:** `Router# show running-config | include traps`	Displays the contents of the currently running configuration file and includes information about enabled traps. • Verifies if the traps have been enabled.

Configuration Examples for IS-IS MIB

Example Enabling and Verifying IS-IS Error Traps

The following example enables all IS-IS error traps:

```
Router(config)# snmp-server enable traps isis
Router(config)# end
```

The **show running-config** command is entered to verify that the traps are enabled:

```
Router# show running-config | include traps
snmp-server enable traps isis
```

Example Enabling and Verifying IS-IS State Change Traps

The following example shows how to enable the ciiDatabaseOverload and ciiManualAddressDrops traps:

```
Router(config)# snmp-server enable traps isis state-change database-overload
Router(config)# snmp-server enable traps isis errors manual-address-drop
Router(config)# end
```

The **show running-config** command is entered to verify that these traps are enabled:

```
Router# show running-config | include traps
snmp-server enable traps isis state-change database-overload
snmp-server enable traps isis errors manual-address-drop
```

Where to Go Next

To configure features to improve IS-IS network convergence times and scalability, complete the optional tasks in one or more of the following modules:

- "Setting Best Practice Parameters for IS-IS Fast Convergence"
- "Reducing Failure Detection Times in IS-IS Networks"
- "Reducing Alternate-Path Calculation Times in IS-IS Networks"

Additional References

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of Cisco IS-IS conceptual information with links to all the individual IS-IS modules	"Integrated IS-IS Routing Protocol Overview"
SNMP configuration	"Configuring SNMP Support" section of the *Cisco IOS XE Network Management Configuration Guide*, Release 2
SNMP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
CISCO-IETF-ISIS-MIB.my	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
IETF draft draft-ietf-isis-wg-mib-16.txt	*Management Information Base for IS-IS*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IS-IS MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 162

IS-IS Support for an IS-IS Instance per VRF for IP

This feature provides multiple VRF-aware IS-IS instances. The VRF functionality allows Internet service providers (ISPs) to separate routing protocol information and propagate it to the appropriate routing table and network neighbors. Using one router with VRF functionality is more cost-effective than using separate routers to separate and forward the routing information.

Prerequisites for IS-IS Support for an IS-IS Instance per VRF for IP

- It is presumed that you are running IS-IS on your network.
- The VRF configuration is a prerequisite to associating an IS-IS instance with that specific VRF. However, the VRF configuration is independent of associating it with IS-IS or any other routing protocol. An IS-IS instance cannot be referred to as being VRF-aware until it has been associated with a particular VRF.

Restrictions for IS-IS Support for an IS-IS Instance per VRF for IP

Support for IS-IS VRF is provided only for IPv4.

When you configure the IS-IS Support for an IS-IS Instance per VRF for IP feature, you must comply with the following nine best-practice guidelines:

- IS-IS instances running Connectionless Network Services (CLNS) must have the same system ID.
- An IS-IS instance that is running CLNS or IPv6 cannot be associated with a VRF.

- You can configure only one IS-IS instance to run both CLNS and IP.
- IS-IS instances within the same VRF must have unique system IDs, although IS-IS instances located in separate VRFs can have the same system ID.
- You can associate an IS-IS instance with only one VRF.
- You can configure the **passive-interface default** command only on one IS-IS instance per VRF.
- Redistribution is allowed only within the same VRF.
- You can enable only one IS-IS instance per interface.
- An interface can belong to an IS-IS instance only if it is associated with the same VRF.

Note If you are using LDP, you cannot use the **route-target** command when configuring a VRF. The router will use BGP for Multiprotocol Label Switching (MPLS) labels.

Information About IS-IS Support for an IS-IS Instance per VRF for IP

VRF-Aware IS-IS

You can configure IS-IS to be VPN routing and forwarding (VRF)-aware. A VRF consists of an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included in the routing table.

IS-IS Support for an IS-IS Instance per VRF for IP Feature Operation

ISPs have the capability to create multiple VRF-aware IS-IS instances that run on one router, rather than requiring duplicate hardware. IS-IS can be enabled to be VRF-aware, and ISPs can use multiple VRF-aware IS-IS instances to separate customer data while propagating the information to appropriate service providers.

For example, an ISP can create three VRFs--VRF First, VRF Second, and VRF Third--to represent three separate customers. A VRF-aware IS-IS instance is created and associated with each VRF: tagFIRST, tagSECOND, and tagTHIRD. Each instance will have its own routing process, IS-IS database, and routing table, and will calculate its own shortest path first (SPF) tree.

How to Configure IS-IS Support for an IS-IS Instance per VRF for IP

Creating a VRF

Before you begin

- It is presumed that you have IS-IS running on your network.
- If CEF is not enabled by default on your platform, you will need to enable CEF in order to associate interfaces with VRF-aware IS-IS instances.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **ip vrf** *vrf-name*
5. **rd** *route-distinguisher*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** [**distributed**] **Example:** `Device(config)# ip cef distributed`	Enables CEF on the Route Processor card. • If CEF is not enabled by default on your particular platform, you must configure it with the **ip cef** command.
Step 4	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf first`	Configures a VRF routing table, and enters VRF configuration mode.

	Command or Action	Purpose
Step 5	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.

Attaching an Interface to the VRF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vrffirst`	Associates a VPN routing and forwarding instance (VRF) with an interface or subinterface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Creating VRF-Aware IS-IS Instances

Prerequisites

Before you create VRF-aware IS-IS instances, you need to enable IP routing on the router.

Note Only one instance within the VRF can be configured as the passive interface default.

Creating a VRF-Aware IS-IS Instance in Interface Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [*secondary*]
5. **ip router isis** *process-tag*
6. **no shutdown**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/2/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [*secondary*] **Example:** `Device(config-if)# ip address 172.16.11.1 255.255.255.255`	Sets a primary or secondary IP address for an interface.
Step 5	**ip router isis** *process-tag* **Example:**	Configures an IS-IS routing process for IP on an interface and attaches a tag to the routing process.

	Command or Action	Purpose
	`Device(config-if)# ip router isis tagfirst`	**Note** The configuration of the interface-mode **ip router isis** command will overwrite the prior configuration on that interface, but only if the new configuration is attempting to change the interface ownership to a different instance that is in the same VRF as the currently configured owner instance. The configuration will be rejected if the attempted change is between two instances that are associated with different VRFs.
Step 6	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts a disabled interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode.

Creating a VRF-Aware IS-IS Instance in Router Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *process-tag*
4. **vrf** *vrf-name*
5. **net** *network-entity-title*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *process-tag* **Example:**	Enables the IS-IS routing protocol, specifies an IS-IS process, and enters router configuration mode.

	Command or Action	Purpose
	`Device(config-if)# router isis tagFirst`	• It is presumed that the VRF named First was previously created.
Step 4	**vrf** *vrf-name* **Example:** `Device(config-router)# vrf first`	Associates an IS-IS instance with a VRF.
Step 5	**net** *network-entity-title* **Example:** `Device(config-router)# net 49.000b.0000.0001.0002.00`	Configures an IS-IS NET for a CLNS routing process.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode.

Configuration Examples for IS-IS Support for an IS-IS Instance per VRF for IP

Example Configuring Multiple VRF-Aware IS-IS Instances

In the following example, the VRF Second is created and an IS-IS instance is created explicitly by entering the **router isis** command on the router:

```
Device(config)# ip cef distributed
Device(config)# ip routing
Device(config)# ip vrf Second
Device(config-vrf)# rd 1:1
Device(config-if)# router isis tagSecond
Device(config-router)# vrf Second
Device(config-router)# net 49.000b.0000.0001.0002.00
```

The VRF Third is created and a VRF-aware IS-IS instance is automatically created when the **ip router isis** command is entered:

```
Device(config)# ip vrf Third
Device(config-vrf)# rd 1:1
Device(config-if)# interface GigabitEthernet0/2/0
Device(config-if)# ip vrf forwarding Third
Device(config-if)# ip address 172.16.10.1 255.255.255.0
Device(config-if)# ip router isis tagThird
Device(config-if)# no shutdown
```

A new IS-IS instance with the process tag tagThird will automatically be created and associated with the VRF Third. When the **show running-config** command is entered, the following information for the new IS-IS instance will be displayed:

```
Device# show running-config
Building configuration...
.
.
.
router isis tagThird
 vrf Third
Device(config)# router isis tagThird
Device(config-router)# net 49.000b.0000.0001.0001.00
```

The following sample output verifies information for the VRF-aware IS-IS instances that were created in the previous examples:

```
Device# show isis tagThird topology
Tag tagThird:
IS-IS paths to level-2 routers
System Id             Metric  Next-Hop                Interface   SNPA
router-02             10      router-02               GE4/3/0     0010.0ddc.e00b
router-03             10      router-03               GE0/2/0     0006.0e03.0c45
router-04             10      router-04               GE4/0/0     000a.f3c3.1c70
.                             router-04               GE4/1/0     000a.f3c3.1c71
.
.
.
Device# show clns tagSecond neighbors
Tag tagSecond:
System Id      Interface   SNPA                State  Holdtime  Type Protocol
router-03      GE0/2/0     00d0.2b7f.9502      Up     9         L2   IS-IS
router-03      PO2/2/0     DLCI 211            Up     27        L2   IS-IS
router-02      PO2/0/0     DLCI 131            Up     29        L2   IS-IS
router-11      GE0/4/0     000e.d79d.7920      Up     7         L2   IS-IS
router-11      GE0/5/0     000e.d79d.7921      Up     8         L2   IS-IS
router-11      PO3/2/0     DLCI 451            Up     24        L2   IS-IS
.
.
.
Device# show isis tagThird database level-2
Tag tagThird:
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
router-01.00-00       0x0000000A    0x5E73       914               0/0/0
router-01.03-00       0x00000001    0x8E41       894               0/0/0
router-01.04-00       0x00000001    0x8747       894               0/0/0
router-03.00-00     * 0x00000005    0x55AD       727               0/0/0
router-03.02-00     * 0x00000001    0x3B97       727               0/0/0
router-02.00-00       0x00000004    0xC1FB       993               0/0/0
router-02.01-00       0x00000001    0x448D       814               0/0/0
router-04.00-00       0x00000004    0x76D0       892               0/0/0
Device# show isis tagThird database level-1
Tag tagThird:
IS-IS Level-1 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
router-03.00-00      * 0x0000000B   0xBDF6        1005              1/0/0
router-03.02-00      * 0x00000001   0xC473        940               0/0/0
router-07.00-00        0x00000006   0x403A        940               0/0/0
Device# show clns tagSecond protocol
IS-IS Router: tagSecond
  System Id: 0000.0001.0002.00  IS-Type: level-2-only
```

```
  Manual area address(es):
        49.000b
  Routing for area address(es):
        49.000b
  Interfaces supported by IS-IS:
        GigabitEthernet4/1/0 - IP
        GigabitEthernet4/0/0 - IP
        GigabitEthernet4/3/0 - IP
  Redistributing:
    static
  Distance: 110
  RRR level: none
  Generate narrow metrics: level-1-2
  Accept narrow metrics:   level-1-2
  Generate wide metrics:   none
  Accept wide metrics:     none
Device# show clns tagThird protocol
IS-IS Router: tagThird
  System Id: 0000.0001.0001.00  IS-Type: level-1-2
  Manual area address(es):
        49.000b
  Routing for area address(es):
        49.000b
  Interfaces supported by IS-IS:
        POS2/2/0 - IP
        GigabitEthernet0/2/0 - IP
        GigabitEthernet0/4/0 - IP
        POS2/0/0 - IP
        GigabitEthernet0/5/0 - IP
        POS3/2/0 - IP
  Redistributing:
    static
  Distance: 110
  RRR level: none
  Generate narrow metrics: none
  Accept narrow metrics:   none
  Generate wide metrics:   level-1-2
  Accept wide metrics:     level-1-2
```

Example Creating an IS-IS Instance Without a Process Tag

In the following example, an IS-IS instance was created without the optional process tag. When an IS-IS instance is created without the optional process tag, you can display its information by entering the commands such as **show clns protocol** with "null" specified for the *process-tag* argument.

```
Device(config)# router isis
Device(config-router)# vrf first
Device(config-router)# net 49.000b.0000.0001.ffff.00
Device(config-router)# is-type level-1
Device(config)# interface POS 6/1/0
Device(config-if)# ip vrf forwarding first
Device(config-if)# ip address 172.16.2.1 255.255.255.0
Device(config-if)# ip router isis
Device(config-if)# no shutdown
```

Because the IS-IS instance is created without the optional process tag, its information is displayed when the **show clns protocol** command is entered with "**null" specified for the** *process-tag* **argument**:

```
Device# show clns null protocol
IS-IS Router: <Null Tag>
```

```
System Id: 0000.0001.FFFF.00  IS-Type: level-1
Manual area address(es):
      49.000b
Routing for area address(es):
      49.000b
Interfaces supported by IS-IS:
      POS6/1/0 - IP
Redistributing:
  static
Distance: 110
RRR level: none
Generate narrow metrics: level-1-2
Accept narrow metrics:   level-1-2
Generate wide metrics:   none
Accept wide metrics:     none
```

Example Redistributing Routes from an IS-IS Instance

In the following sample configuration, routes have been redistributed from the IS-IS instance "null" into the IS-IS instance named tagBLUE. Routes from an OSPF process in VRF Blue have been redistributed into the IS-IS instance named tagBLUE.

```
Device(config)# router isis tagBLUE
Device(config-router)# redistribute isis null ip metric 10 route-map isisMAP1
Device(config-router)# redistribute ospf 1 vrf BLUE metric 1 metric-type external
 level-1-2
.
.
.
Device(config)# route-map isisMAP1 permit 10
Device(config-route-map)# match route-type level-2 level-1
Device(config-route-map)# set level level-2
```

Example Changing the Interface Ownership

In the following sample configuration, POS interface 6/1/0 was originally enabled for IS-IS IP routing for a "null" instance that does not have a process tag, which is in vrfSecond. The new configuration changes the ownership of POS interface 6/1/0 to another instance tagSecond, which is also in vrfSecond.

Note Note that use of the **ip router isis**command in interface configuration mode will overwrite the prior configuration on that interface, but only if the new configuration is attempting to change the interface ownership to a different instance that is in the same VRF as the currently configured owner instance. The configuration will be rejected if the attempted change is between two instances that are associated with different VRFs.

```
Device(config)# interface POS 6/1/0
Device(config-if)# ip router isis tagSecond
%ISIS: Interface detached from null and to be attached to instance tagBLUE.
```

Additional References

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of Cisco IS-IS conceptual information with links to all the individual IS-IS modules	"Integrated IS-IS Routing Protocol Overview" module
ISO CLNS commands	*Cisco IOS ISO CLNS Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IS-IS Support for an IS-IS Instance per VRF for IP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 163

Overview of IS-IS Fast Convergence

This module provides information about the topics of Intermediate System-to-Intermediate System (IS-IS) fast convergence. The tasks in the modules that follow this overview can help you improve convergence times for IS-IS networks.

Prerequisites for IS-IS Fast Convergence

You should be familiar with the concepts described in the "Integrated IS-IS Routing Protocol Overview" module.

Information About IS-IS Fast Convergence

You should understand the following concepts before you configure any features to improve IS-IS network convergence times:

Network Convergence

Convergence is the process of all routers coming to agreement on optimal routes in a network. When a network event causes routes to become available or unavailable, routers send routing update messages through the network that cause routing algorithms to recalculate optimal routes. Eventually all the routers agree on the routes as well as the network topology. Fast convergence benefits network performance. Routing algorithms that converge slowly may cause temporary routing loops or temporary network unavailability.

The process of network convergence can be divided into three separate stages:

1. Routing change detection: The speed at which a device on the network can detect and react to the failure or modification of one of its own components, or to a topology change caused by the failure or modification of a component on a routing protocol peer.
2. Routing change notification: The speed at which the failure or topology change in the previous stage can be communicated to other devices in the network.

3. Alternate path calculation: The speed at which all devices on the network, having been notified of the failure or topology change, can process the information and calculate an alternate path through which data can flow.

An improvement in any one of these stages provides an improvement in overall convergence. In addition to a basic configuration task that is recommended as a first step in configuring an IS-IS router with best practice parameters for achieving fast convergence, several recommended configuration tasks are grouped according to the stage of network convergence they can improve. For more information, see the following modules:

- "Setting Best Practice Parameters for IS-IS Fast Convergence"
- "Reducing Failure Detection Times in IS-IS Networks"
- "Reducing Link Failure and Topology Change Notification Times in IS-IS Networks"
- "Reducing Alternate-Path Calculation Times in IS-IS Networks"

Design Recommendations for Achieving Faster Network Convergence

A faster processor can provide better performance for network convergence.

Where to Go Next

To configure features to improve IS-IS network convergence times, complete the optional tasks in one or more of the following modules:

- "Setting Best Practice Parameters for IS-IS Fast Convergence"
- "Reducing Failure Detection Times in IS-IS Networks"
- "Reducing Alternate-Path Calculation Times in IS-IS Networks"

To enhance IS-IS network security, see the "Enhancing Security in an IS-IS Network" module.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Related Topic	Document Title
IPv6 Routing: IS-IS Support for IPv6	"*Integrated IS-IS Routing Protocol Overview* " module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overview of IS-IS Fast Convergence

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 164

Setting Best Practice Parameters for IS-IS Fast Convergence

This module describes how to configure an IS-IS router with parameters that are recommended as a basic step to improve network convergence.

Prerequisites for Setting Best Practice Parameters for IS-IS Fast Convergence

- It is assumed that you already have IS-IS running on your network.
- Before performing the tasks in this module, you should be familiar with the concepts described in the "Overview of IS-IS Fast Convergence" module.

Information About Setting Best Practice Parameters for IS-IS Fast Convergence

Information About Increased Scaling of IS-IS Neighbors

How to Set Best Practice Parameters for IS-IS Fast Convergence

Setting Best Practice Parameters for IS-IS Fast Convergence

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [**area-** *tag*]
4. **is-type** [**level-1** | **level-1-2** | **level-2-only**]
5. **metric-style wide** [**transition**] [**level-1** | **level-2** | **level-1-2**]
6. **set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] [**suppress** {**interlevel**| **external**}]
7. **no hello padding**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [**area-** *tag*] **Example:** `Router(config)# router isis`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**is-type** [**level-1**	**level-1-2**	**level-2-only**] **Example:**	Configures the routing level for an instance of the IS-IS routing process.

	Command or Action	Purpose
	`Router(config-router)# is-type level-1`	• It is recommended that IS-IS nodes that operate at a single level be configured as Level 1 to minimize the number of adjacencies, LDSBs, and related SPF and PRC calculations. **Note** You can also set the IS-IS level type on the interface by entering the **isis circuit-type** command.
Step 5	**metric-style wide** [**transition**] [**level-1** | **level-2** | **level-1-2**] **Example:** `Router(config-router)# metric-style wide`	Globally changes the metric value for all IS-IS interfaces. • Wide style metrics are required for prefix tagging.
Step 6	**set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] [**suppress** {**interlevel**| **external**}] **Example:** `Router(config-router)# set-overload-bit on-startup 360`	Configures the router to signal other routers not to use it as an intermediate hop in their shortest path first (SPF) calculations. • Setting the overload bit gives the router enough time to build its BGP and CEF tables prior to the router being used as a transit node.
Step 7	**no hello padding** **Example:** `Router(config-router)# no hello padding`	Disables IS-IS hello padding at the router level. • By default the IS-IS Hello PDUs are padded to the full MTU size, possibly having a negative impact on time-sensitive application traffic that travels across low-bandwidth interfaces or on interface buffer resources when frequent hellos are configured. It is recommended to globally disable hello padding.
Step 8	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuration Examples for Setting Best Practice Parameters for IS-IS Fast Convergence

Example Enabling IS-IS on a Router and Setting Best Practice Parameters for IS-IS Fast Convergence

The following example enables the IS-IS routing protocol on the interfaces for Router A, enables IS-IS on Router A, and configures Router A with the basic commands recommended to optimize IS-IS network convergence.

Router A

```
!
clns routing
process-max-time 50
ip routing protocol purge interface
router isis
 passive-interface Loopback0
 net 49.1962.XXXX.XXXX.XXXX.00
 is-type level-2-only
 ispf level-2
 log-adjacency-changes
 ignore-lsp-errors
 metric-style wide level-2
 external overload signalling !Configure on Cisco 12000 series Internet routers
 set-overload-bit on-startup 180
 max-lsp-lifetime 65535
 lsp-refresh-interval 65000
 spf-interval 5 1 50
 prc-interval 5 1 50
 lsp-gen-interval 5 1 50
 no hello padding
 authentication mode md5 level-2
 authentication key-chain ON
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-2
!
interface GigabitEthernet x/x/x
 negotiation auto
 ip router isis
 mtu 4470
 isis network point-to-point
 isis metric <metric> level-2
 isis circuit-type level-2-only
 isis authentication mode md5 level-2
 isis authentication key-chain ON
 carrier-delay ms 0
 dampening
interface POSx/y/x
 carrier-delay msec 0
 dampening
 ip router isis
 no peer neighbor-route
 isis metric 1 level-2
 isis circuit-type level-2-only
 isis authentication mode md5 level-2
 isis authentication key-chain ON
 pos ais-shut
 pos report lais
 pos report lrdi
 pos report pais
 pos report prdi
 pos report slos
 pos report slof
!
key chain ON
 key 1
  key-string mypassword
```

Where to Go Next

To configure features to improve IS-IS network convergence times, complete the optional tasks in one or more of the following modules:

- "Reducing Failure Detection Times in IS-IS Networks"
- "Reducing Alternate-Path Calculation Times in IS-IS Networks"

Additional References

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of Cisco IS-IS conceptual information	"Overview of IS-IS Fast Convergence"
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Setting Best Practice Parameters for IS-IS Fast Convergence

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 165

Best Practices for Increased Scaling of IS-IS Neighbors

This chapter describes how to increase scaling of neighbors in a hub and spoke deployment using the following configuration options:

- Reducing flooding over parallel peer-to-peer links
- Staggered synchronization of adjacencies after router reload
- Configuring and monitoring IS-IS queue

Before You Begin

- It is assumed that you already have IS-IS running on your network.
- Before performing the tasks in this module, you should be familiar with the concepts described in the "Overview of IS-IS Fast Convergence" module.

Information About Increased Scaling of IS-IS Neighbors

Controlling Flooding Over Parallel Peer-to-Peer Links

In the presence of parallel links to the same neighbor, redundant flooding of link state packet (LSP) updates occur on each link, by default. This when combined with large number of neighbors, exponentially increases the CPU processing time required to synchronize the LSP database. The suppression of flooding over parallel links improves performance at scale with no loss of reliability.

Staggered Synchronization of Adjacencies After Router Reload

The simultaneous boot up of a large number of adjacencies after a reboot or IS-IS process restart can result in overflow of input queues due to receipt of redundant link state information from multiple neighbors. To avoid such situations, it is important to stagger synhronization of adjacencies and also limit the maximum number of adjacencies that are in syncing state. This mechanism also helps in situations where network issues may affect synchronization of adjacency information on some interfaces.

Note Staggered synchronization of adjacencies is only supported on point-to-point interfaces

Setting Up and Monitoring IS-IS Queues

For large scale deployments, when the queue reaches a high watermark, it can lead to dropping of packets. In such a situation, it is important to adjust the IS-IS queue size to handle high loads. Additionally, you can monitor the IOS queue to check:

- Current state of the IS-IS input queue
- Queue size maximum limit
- Highest water mark (maximum size queue reached since last set of counters were cleared)
- Number of drops

How to Configure Increased Scaling of IS-IS Neighbors

Configuring Flooding Reduction For Parallel Links

```
enable
configure terminal
router isis [area-tag]
flood parallel suppression
```

By default, flood suppression functionality is enabled.

Configuring IS-IS Input Queue Size

```
enable
clns queue-depth <size>
```

Configuring Staggered Synchronization of Adjacencies

```
enable
configure terminal
router isis [area-tag]
adjacency stagger <initial> <max>
```

This feature is disabled by default. If staggering of adjacencies is enabled, you can use the **show isis protocol** command to see the active values as well as the *syncing* and *full* count of point-to-point adjacencies.

Monitoring ISIS Queues

You can use the `show clns traffic` command to check:

- Actual queue size
- Queue size maximum limit
- Highest water mark (maximum size queue reached since last set of counters were cleared)
- Number of drops

CHAPTER 166

Reducing Failure Detection Times in IS-IS Networks

This module describes how to customize IS-IS configuration to help you achieve fast convergence in your network. This module describes tasks to optimize how a router that runs IS-IS detects link failures and topology changes, sends important topology change updates to its neighbors, and reacts to the topology change updates that it receives from its neighbors, in order to increase network performance.

Prerequisites for Reducing Failure Detection Times in IS-IS Networks

You should be familiar with the concepts described in the "Overview of IS-IS Fast Convergence" module.

Information About Reducing Failure Detection Times in IS-IS Networks

IP event dampening introduces a configurable exponential delay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network operator to configure a router to automatically identify and selectively dampen a local interface that is flapping, removing it from the network until it becomes stable again. Thus, the network becomes more stable, with a faster convergence time.

Tuning hello parameters should be considered only when the link type does not offer fast enough link failure detection. The standard default values for the hello interval and hello multiplier are 10 seconds and 3 seconds. Therefore, the multiplier times the interval will give a default hold-time of 30 seconds.

Although a slower hello interval saves bandwidth and CPU usage, there are some situations when a faster hello interval is preferred. In the case of a large configuration that uses Traffic Engineering (TE) tunnels, if the TE tunnel uses ISIS as the Interior Gateway Protocol (IGP), and the IP routing process is restarted at the router at the ingress point of the network (headend), then all the TE tunnels get resignaled with the default hello interval. A faster hello interval prevents this resignaling. To configure a faster hello interval, you need to decrease the ISIS hello interval manually using the **isis hello-interval**command.

Configuring a point-to-point adjacency over a broadcast media can improve convergence times of a customer's network because it prevents the system from electing a designated router (DR), prevents flooding from using CSNPs for database synchronization, and simplifies shortest path first (SPF) computations.

Importance of Fast Network Failure Detection

You can customize your IS-IS network to reduce the amount of time it takes for network failures to be discovered. When failures are detected more quickly, networks can react to them sooner and alternate paths can be selected more quickly, speeding up network convergence.

How to Reduce Failure Detection Times in IS-IS Networks

Using IP Event Dampening to Decrease Failure Detection Times

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dampening** [*half-life-period reuse-threshold*] [*suppress-threshold max-suppress-time* [*restart-penalty*]]
5. **end**
6. **show dampening interface**
7. **show interface dampening**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters interface configuration mode.

	Command or Action	Purpose
Step 4	**dampening** [*half-life-period reuse-threshold*] [*suppress-threshold max-suppress-time* [*restart-penalty*]] **Example:** `Device(config-if)# dampening`	Enables interface dampening. • Entering the **dampening** command without any keywords or arguments enables interface dampening with the default configuration parameters. **Note** The default values for the *half-life-period*, *reuse-threshold*, *suppress-threshold*, *max-suppress-time*, and *restart-penalty* arguments are 5, 1000, 2000, 20, and 2000, respectively. • When the timer for the *restart-penalty* argument is manually configured, the values must be manually entered for all arguments.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show dampening interface** **Example:** `Device# show dampening interface`	Displays a summary of dampened interfaces.
Step 7	**show interface dampening** **Example:** `Device# show interface dampening`	Displays dampened interfaces on the local router.

Tuning IS-IS Hello Parameters to Decrease Link Failure Detection Times

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**]
5. **isis hello-multiplier** *multiplier* [**level-1** | **level-2**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:**	Configures an interface type and enters interface configuration mode.
Step 4	**isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**] **Example:** `Device(config-if)# isis hello-interval 5 level-1`	Specifies the length of time between the sending of IS-IS hello PDUs. • The default value is 10. The hello interval multiplied by the hello multiplier equals the hold time. If the minimal keyword is specified, the hold time is 1 second and the system computes the hello interval based on the hello multiplier. • The hello interval can be configured independently for Level 1 and Level 2, except on serial point-to-point interfaces. (Because only a single type of hello PDU is sent on serial links, it is independent of Level 1 or Level 2.) The level-1 and level-2 keywords are used on X.25, SMDS, and Frame Relay multiaccess networks or LAN interfaces. **Note** A faster hello interval gives faster convergence, but increases bandwidth and CPU usage. It might also add to instability in the network, due to false failure detection events. A slower hello interval saves bandwidth and CPU. Especially when used in combination with a higher hello multiplier, this configuration may increase overall network stability, but has typical slower network convergence as a consequence.
Step 5	**isis hello-multiplier** *multiplier* [**level-1** | **level-2**] **Example:** `Device(config-if)# isis hello-multiplier 6 level-1`	Specifies the number of IS-IS hello PDUs a neighbor must miss before the router should declare the adjacency as down. • The default value is 3. A multiplier value of 1 is very aggressive--we recommend a value of at least 3.
Step 6	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

Configuring an IS-IS Point-to-Point Adjacency over Broadcast Media

Perform this task for IS-IS networks that consist of only two networking devices connected to broadcast media. Such networks are usually configured as a point-to-point link rather than a broadcast link.

Note Having a multipoint interface instead of a point-to-point interface will cause the creation of a pseudonode on the network. The addition of the pseudonode means that the router must retain information about it. To decrease the size of the topology database of the router, thereby reducing the memory requirement of the router and increasing the efficiency of the SPF calculation since there is one less node involved, configure point-to-point interfaces when possible.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **isis network point-to-point**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:**	Configures an interface type and enters interface configuration mode.
Step 4	**isis network point-to-point** **Example:** `Device(config-if)# isis network point-to-point`	Configures a network of only two networking devices that use broadcast media and the integrated IS-IS routing protocol to function as a point-to-point link instead of a broadcast link.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Monitoring IS-IS Network Convergence Time

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isis display delimiter** [**return** *count* | **character** *count*]
4. **exit**
5. **show isis database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
6. **show isis** [*process-tag*] **route**
7. **show isis spf-log**
8. **show isis** [*process-tag*] **topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**isis display delimiter** [**return** *count*	**character** *count*] **Example:** `Device(config)# isis display delimiter return 2`	Makes output from multiarea displays easier to read by specifying the delimiter to use to separate displays of information.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show isis database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:**	Displays the IS-IS link-state database.

	Command or Action	Purpose
	`Device# show isis database detail`	
Step 6	**show isis** [*process-tag*] **route** **Example:** `Device# show isis financetag route`	Displays the IS-IS Level 1 forwarding table for IS-IS learned routes.
Step 7	**show isis spf-log** **Example:** `Device# show isis spf-log`	Displays how often and why the router has run a full SPF calculation.
Step 8	**show isis** [*process-tag*] **topology** **Example:** `Device# show isis financetag topology`	Displays a list of all connected routers in all areas. • If a process tag is specified, output is limited to the specified routing process. When "null" is specified for the process tag, output is displayed only for the router process that has no tag specified. If a process tag is not specified, output is displayed for all processes.

Configuration Examples for Reducing Failure Detection Times in IS-IS Networks

Example Configuring IS-IS to Achieve Fast Convergence by Reducing Failure Detection Times

The following example configures Ethernet interface 0/0 to use IP event dampening, setting the half life to 30 seconds, the reuse threshold to 1500, the suppress threshold to 10,000, and the maximum suppress time to 120 seconds. The IS-IS hello parameters have also been tuned for more rapid failure detection

```
enable
configure terminal
interface Ethernet 0/0
 dampening 30 1500 10000 120
 isis hello-interval minimal
 isis hello-multiplier 3
```

Where to Go Next

To configure additional features to improve IS-IS network convergence times, complete the optional tasks in one or more of the following modules:

- "Setting Best Practice Parameters for IS-IS Fast Convergence"

- "Reducing Link Failure and Topology Change Notification Times in IS-IS Networks"
- "Reducing Alternate-Path Calculation Times in IS-IS Networks"

Additional References

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of Cisco IS-IS conceptual information with links to all the individual IS-IS modules	"Integrated IS-IS Routing Protocol Overview"

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Reducing Failure Detection Times in IS-IS Networks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 167

IPv6 Routing: IS-IS Multitopology Support for IPv6

IS-IS multitopology support for IPv6 allows IS-IS to maintain a set of independent topologies within a single area or domain.

IPv6 Routing: IS-IS Multitopology Support for IPv6

IS-IS Enhancements for IPv6

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and OSI routes. Extensions to the IS-IS command-line interface (CLI) allow configuration of IPv6-specific parameters. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4.

IS-IS in IPv6 supports either single-topology mode or multiple topology mode.

IS-IS Multitopology Support for IPv6

IS-IS multitopology support for IPv6 allows IS-IS to maintain a set of independent topologies within a single area or domain. This mode removes the restriction that all interfaces on which IS-IS is configured must support the identical set of network address families. It also removes the restriction that all routers in the IS-IS area (for Level 1 routing) or domain (for Level 2 routing) must support the identical set of network layer address families. Because multiple SPFs are performed, one for each configured topology, it is sufficient that connectivity exists among a subset of the routers in the area or domain for a given network address family to be routable.

You can use the **isis ipv6 metric** command to configure different metrics on an interface for IPv6 and IPv4.

When multitopology support for IPv6 is used, use the **metric-style wide**command to configure IS-IS to use new-style TLVs because TLVs used to advertise IPv6 information in link-state packets (LSPs) are defined to use only extended metrics.

Transition from Single-Topology to Multitopology Support for IPv6

All routers in the area or domain must use the same type of IPv6 support, either single-topology or multitopology. A router operating in multitopology mode will not recognize the ability of the single-topology mode router to support IPv6 traffic, which will lead to holes in the IPv6 topology. To transition from single-topology support to the more flexible multitopology support, a multitopology transition mode is provided.

The multitopology transition mode allows a network operating in single-topology IS-IS IPv6 support mode to continue to work while upgrading routers to include multitopology IS-IS IPv6 support. While in transition mode, both types of TLVs (single-topology and multitopology) are sent in LSPs for all configured IPv6 addresses, but the router continues to operate in single-topology mode (that is, the topological restrictions of the single-topology mode are still in effect). After all routers in the area or domain have been upgraded to support multitopology IPv6 and are operating in transition mode, transition mode can be removed from the configuration. Once all routers in the area or domain are operating in multitopology IPv6 mode, the topological restrictions of single-topology mode are no longer in effect.

How to Configure IPv6 Routing: IS-IS Multitopology Support for IPv6

Configuring Multitopology IS-IS for IPv6

When multitopology IS-IS for IPv6 is configured, the **transition** keyword allows a user who is working with the single-topology SPF mode of IS-IS IPv6 to continue to work while upgrading to multitopology IS-IS. After every router is configured with the **transition** keyword, users can remove the **transition** keyword on each router. When transition mode is not enabled, IPv6 connectivity between routers operating in single-topology mode and routers operating in multitopology mode is not possible.

You can continue to use the existing IPv6 topology while upgrading to multitopology IS-IS. The optional **isis ipv6 metric** command allows you to differentiate between link costs for IPv6 and IPv4 traffic when operating in multitopology mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **metric-style wide** [**transition**] [**level-1** | **level-2** | **level-1-2**
5. **address-family ipv6** [**unicast** | **multicast**]
6. **multi-topology** [**transition**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Device(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**metric-style wide** [**transition**] [**level-1** | **level-2** | **level-1-2** **Example:** `Device(config-router)# metric-style wide level-1`	Configures a router running IS-IS to generate and accept only new-style TLVs.
Step 5	**address-family ipv6** [**unicast** | **multicast**] **Example:** `Device(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 6	**multi-topology** [**transition**] **Example:** `Device(config-router-af)# multi-topology`	Enables multitopology IS-IS for IPv6. • The optional **transition** keyword allows an IS-IS IPv6 user to continue to use single-topology mode while upgrading to multitopology mode.

Customizing IPv6 IS-IS

Perform this task to configure a new administrative distance for IPv6 IS-IS, configure the maximum number of equal-cost paths that IPv6 IS-IS will support, configure summary prefixes for IPv6 IS-IS, and configure an IS-IS instance to advertise the default IPv6 route (::/0). It also explains how to configure the hold-down period between partial route calculations (PRCs) and how often Cisco IOS XE software performs the SPF calculation when using multitopology IS-IS.

You can customize IS-IS multitopology for IPv6 for your network, but you likely will not need to do so. The defaults for this feature are set to meet the requirements of most customers and features. If you change the defaults, refer to the IPv4 configuration guide and the IPv6 command reference to find the appropriate syntax.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast** | **multicast**]

5. **default-information originate** [**route-map** *map-name*]
6. **distance** *value*
7. **maximum-paths** *number-paths*
8. **summary-prefix** *ipv6-prefix prefix-length* **level-1** | **level-1-2**| **level-2**]
9. **prc-interval** *seconds* [*initial-wait*] [*secondary-wait*
10. **spf-interval** [**level-1** | **level-2**] *seconds initial-wait*] [*secondary-wait*
11. **exit**
12. **interface** *type number*
13. **isis ipv6 metric** *metric-value* [**level-1** | **level-2** | **level-1-2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** *area-tag* **Example:** `Router(config)# router isis area2`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	**address-family ipv6** [**unicast** | **multicast**] **Example:** `Router(config-router)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. • The **unicast** keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the **unicast** keyword is not specified with the **address-family ipv6** command.
Step 5	**default-information originate** [**route-map** *map-name*] **Example:** `Router(config-router-af)# default-information originate`	(Optional) Injects a default IPv6 route into an IS-IS routing domain. • The **route-map** keyword and *map-name* argument specify the conditions under which the IPv6 default route is advertised. • If the **route map** keyword is omitted, then the IPv6 default route will be unconditionally advertised at Level 2.

	Command or Action	Purpose
Step 6	**distance** *value* **Example:** `Router(config-router-af)# distance 90`	(Optional) Defines an administrative distance for IPv6 IS-IS routes in the IPv6 routing table. • The *value*argument is an integer from 10 to 254. (The values 0 to 9 are reserved for internal use).
Step 7	**maximum-paths** *number-paths* **Example:** `Router(config-router-af)# maximum-paths 3`	(Optional) Defines the maximum number of equal-cost routes that IPv6 IS-IS can support. • This command also supports IPv6 Border Gateway Protocol (BGP) and Routing Information Protocol (RIP). • The *number-paths* argument is an integer from 1 to 64. The default for BGP is one path; the default for IS-IS and RIP is 16 paths.
Step 8	**summary-prefix** *ipv6-prefix prefix-length* **level-1** | **level-1-2**| **level-2**] **Example:** `Router(config-router-af)# summary-prefix 2001:DB8::/24`	(Optional) Allows a Level 1-2 router to summarize Level 1 prefixes at Level 2, instead of advertising the Level 1 prefixes directly when the router advertises the summary. • The *ipv6-prefix* argument in the **summary-prefix** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. • The *prefix-length* argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 9	**prc-interval** *seconds* [*initial-wait*] [*secondary-wait* **Example:** `Router(config-router-af)# prc-interval 20`	(Optional) Configures the hold-down period between PRCs for multitopology IS-IS for IPv6.
Step 10	**spf-interval** [**level-1** | **level-2**] *seconds initial-wait*] [*secondary-wait* **Example:** `Router(config-router-af)# spf-interval 30`	(Optional) Configures how often Cisco IOS XE software performs the SPF calculation for multitopology IS-IS for IPv6.
Step 11	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode, and returns the router to router configuration mode. • Repeat this step to exit router configuration mode and return the router to global configuration mode.

	Command or Action	Purpose
Step 12	**interface** *type number* **Example:** `Router(config-router)# interface GigabitEthernet 0/0/1`	Specifies the interface type and number, and enters interface configuration mode.
Step 13	**isis ipv6 metric** *metric-value* [**level-1** | **level-2** | **level-1-2** **Example:** `Router(config-if)# isis ipv6 metric 20`	(Optional) Configures the value of an multitopology IS-IS for IPv6 metric.

Verifying IPv6 IS-IS Configuration and Operation

SUMMARY STEPS

1. **enable**
2. **show ipv6 protocols** [**summary**]
3. **show isis** [*process-tag*] [**ipv6** | *****] **topology**
4. **show clns** [*process-tag*] **neighbors** *interface-type interface-number*] [**area**] [**detail**]
5. **show clns** *area-tag* **is-neighbors** [*type number*] [**detail**]
6. **show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
7. **show isis ipv6 rib** [*ipv6-prefix*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 protocols** [**summary**] **Example:** `Device# show ipv6 protocols`	Displays the parameters and current state of the active IPv6 routing processes.
Step 3	**show isis** [*process-tag*] [**ipv6** | *****] **topology** **Example:** `Device# show isis topology`	Displays a list of all connected routers running IS-IS in all areas.
Step 4	**show clns** [*process-tag*] **neighbors** *interface-type interface-number*] [**area**] [**detail**] **Example:**	Displays end system (ES), intermediate system (IS), and multitopology IS-IS (M-ISIS) neighbors.

	Command or Action	Purpose
	`Device# show clns neighbors detail`	
Step 5	**show clns** *area-tag* **is-neighbors** [*type number*] [**detail**] **Example:** `Device# show clns is-neighbors detail`	Displays IS-IS adjacency information for IS-IS neighbors. • Use the **detail** keyword to display the IPv6 link-local addresses of the neighbors.
Step 6	**show isis** [*process-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:** `Device# show isis database detail`	Displays the IS-IS link-state database. • In this example, the contents of each LSP are displayed using the **detail** keyword.
Step 7	**show isis ipv6 rib** [*ipv6-prefix*] **Example:** `Device# show isis ipv6 rib`	Displays the IPv6 local RIB.

Configuration Examples for IPv6 Routing: IS-IS Multitopology Support for IPv6

Example: Configuring the IS-IS IPv6 Metric for Multitopology IS-IS

The following example sets the value of an IS-IS IPv6 metric to 20:

```
interface GigabitEthernet 0/0/1
 isis ipv6 metric 20
```

Example: Configuring IS-IS for IPv6

In the following example, output information about the parameters and current state of that active IPv6 routing processes is displayed using the **show ipv6 protocols**command:

```
Device# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "isis"
  Interfaces:
    GigabitEthernet0/0/3
    GigabitEthernet0/0/1
    Serial1/0/1
    Loopback1 (Passive)
    Loopback2 (Passive)
    Loopback3 (Passive)
    Loopback4 (Passive)
```

```
    Loopback5 (Passive)
  Redistribution:
    Redistributing protocol static at level 1
  Address Summarization:
    L2: 2001:DB8:33::/16  advertised with metric 0
    L2: 2001:DB8:44::/16  advertised with metric 20
    L2: 2001:DB8:66::/16  advertised with metric 10
    L2: 2001:DB8:77::/16  advertised with metric 10
```

In the following example, output information about all connected routers running IS-IS in all areas is displayed using the **show isis topology**command:

```
Device# show isis topology
IS-IS paths to level-1 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.000C
0000.0000.000D  20      0000.0000.00AA  Se1/0/1         *HDLC*
0000.0000.000F  10      0000.0000.000F  GE0/0/1         0050.e2e5.d01d
0000.0000.00AA  10      0000.0000.00AA  Se1/0/1         *HDLC*
IS-IS paths to level-2 routers
System Id       Metric  Next-Hop        Interface       SNPA
0000.0000.000A  10      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000B  20      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000C  --
0000.0000.000D  30      0000.0000.000A  GE0/0/3         0010.f68d.f063
0000.0000.000E  30      0000.0000.000A  GE0/0/3         0010.f68d.f063
```

In the following example, output information to confirm that the local router has formed all the necessary IS-IS adjacencies with other IS-IS neighbors is displayed using the **show clns is-neighbors**command. To display the IPv6 link-local addresses of the neighbors, specify the **detail** keyword.

```
Device# show clns is-neighbors detail
System Id      Interface   State  Type Priority  Circuit Id         Format
0000.0000.00AA Se1/0/1     Up     L1   0         00                 Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::YYYY:D37C:C854:5
  Uptime: 17:21:38
0000.0000.000F Et0/0/1     Up     L1   64        0000.0000.000C.02  Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::XXXX:E2FF:FEE5:D01D
  Uptime: 17:21:41
0000.0000.000A Et0/0/3     Up     L2   64        0000.0000.000C.01  Phase V
  Area Address(es): 49.000b
  IPv6 Address(es): FE80::ZZZZ:F6FF:FE8D:F063
  Uptime: 17:22:06
```

In the following example, detailed output information that displays both end system (ES) and intermediate system (IS) neighbors is displayed using the **show clns neighbors** command with the **detail** keyword.

```
Device# show clns neighbors detail
System Id          Interface    SNPA            State  Holdtime  Type Protocol
0000.0000.0007     GE3/3        aa00.0400.6408  UP     26        L1   IS-IS
Area Address(es): 20
IP Address(es): 172.16.0.42*
Uptime: 00:21:49
0000.0C00.0C35     GE3/2        0000.0c00.0c36  Up     91        L1   IS-IS
Area Address(es): 20
IP Address(es): 192.168.0.42*
Uptime: 00:21:52
```

```
0800.2B16.24EA     GE3/3        aa00.0400.2d05  Up     27         L1   M-ISIS
Area Address(es): 20
IP Address(es): 192.168.0.42*
IPv6 Address(es): FE80::2B0:8EFF:FE31:EC57
Uptime: 00:00:27
0800.2B14.060E     GE3/2        aa00.0400.9205  Up     8          L1   IS-IS
Area Address(es): 20
IP Address(es): 192.168.0.30*
Uptime: 00:21:52
```

In the following example, detailed output information about LSPs received from other routers and the IPv6 prefixes they are advertising is displayed using the **show isis database**command with the **detail** keyword specified:

```
Device# show isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C00.0C35.00-00  0x0000000C   0x5696        325           0/0/0
  Area Address: 47.0004.004D.0001
  Area Address: 39.0001
  Metric: 10   IS 0000.0C00.62E6.03
  Metric: 0    ES 0000.0C00.0C35
 --More--
0000.0C00.40AF.00-00* 0x00000009   0x8452        608           1/0/0
  Area Address: 47.0004.004D.0001
  Topology: IPv4 (0x0) IPv6 (0x2)
  NLPID: 0xCC 0x8E
  IP Address: 172.16.21.49
  Metric: 10   IS 0800.2B16.24EA.01
  Metric: 10   IS 0000.0C00.62E6.03
  Metric: 0    ES 0000.0C00.40AF
  IPv6 Address: 2001:DB8::/32
  Metric: 10   IPv6 (MT-IPv6) 2001:DB8::/64
  Metric: 5    IS-Extended cisco.03
  Metric: 10   IS-Extended cisco1.03
  Metric: 10    IS (MT-IPv6) cisco.03
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
0000.0000.000A.00-00  0x00000059   0x378A        949               0/0/0
  Area Address: 49.000b
  NLPID:        0x8E
  IPv6 Address: 2001:DB8:1:1:1:1:1:1
  Metric: 10         IPv6 2001:DB8:2:YYYY::/64
  Metric: 10         IPv6 2001:DB8:3:YYYY::/64
  Metric: 10         IPv6 2001:DB8:2:YYYY::/64
  Metric: 10         IS-Extended 0000.0000.000A.01
  Metric: 10         IS-Extended 0000.0000.000B.00
  Metric: 10         IS-Extended 0000.0000.000C.01
  Metric: 0          IPv6 11:1:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:2:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:3:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:4:YYYY:1:1:1:1:1/128
  Metric: 0          IPv6 11:5:YYYY:1:1:1:1:1/128
0000.0000.000A.01-00  0x00000050   0xB0AF        491               0/0/0
  Metric: 0          IS-Extended 0000.0000.000A.00
  Metric: 0          IS-Extended 0000.0000.000B.00
```

The following example shows output from the **show isis ipv6 rib** command. An asterisk (*) indicates prefixes that have been installed in the primary IPv6 RIB as IS-IS routes. Following each prefix is a list of all paths in order of preference, with optimal paths listed first and suboptimal paths listed after optimal paths.

```
Device# show isis ipv6 rib

IS-IS IPv6 process "", local RIB
  2001:DB8:88:1::/64
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L2  metric 20 LSP [3/7]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L2  metric 20 LSP [3/7]
* 2001:DB8:1357:1::/64
    via FE80::202:7DFF:FE1A:9471/GigabitEthernet2/1/0, type L2  metric 10 LSP [4/9]
* 2001:DB8:45A::/64
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L1  metric 20 LSP [C/6]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L1  metric 20 LSP [C/6]
    via FE80::210:7BFF:FEC2:ACC9/GigabitEthernet2/0/0, type L2  metric 20 LSP [3/7]
    via FE80::210:7BFF:FEC2:ACCC/GigabitEthernet2/1/0, type L2  metric 20 LSP [3/7]
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
IPv6 Routing: IS-IS Multitopology Support for IPv6	" *Reducing Link Failure and Topology Change Notification Times in IS-IS Networks* " module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: IS-IS Multitopology Support for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 168

Reducing Link Failure and Topology Change Notification Times in IS-IS Networks

The tasks in this module explain how to customize Intermediate System-to-Intermediate System (IS-IS) to reduce the amount of time required for routers to send link failure and topology change information to neighbors. You can adjust the IS-IS timers and thereby decrease the time required for a device to send routing updates.

Prerequisites for Reducing Link Failure and Topology Change Notification Times in IS-IS Networks

Before performing the tasks in this module, you should be familiar with the concepts described in the ""Overview of IS-IS Fast Convergence" module.

Information About Reducing Link Failure and Topology Change Notification Times in IS-IS Networks

IS-IS LSP Generation Interval and Lifetime

If you increase the link-state Protocol Data Unit (PDU) LSP tuning values to their maximum, flooding will be significantly reduced, as will resource consumption by the flooding mechanism. The maximum period a router is allowed to wait before regenerating its LSP is approximately 18.7 hours.

IS-IS Throttling Timers That Affect Fast Convergence

You can configure IS-IS to react more rapidly to isolated events that are likely to be real link failures and to react more stably to frequent events that are unlikely to be actual link failures. The convergence speed and stability of IS-IS is affected by the values that you set for various throttling timers. The throttling timers impose a trade-off between reaction time to external events and the amount of resources dedicated to maintaining the information in the Routing Information Base (RIB). You should become familiar with the following.

IS-IS PDUs

IS-IS encapsulates data into a data-link PDU. There are four different PDU types and each can be Level 1 or Level 2:

- **LSP** --An LSP is a PDU that is sent between two IS-IS neighbors. The LSP contains information about neighbors and path costs, including adjacencies to neighbors, connected IP prefixes, Open Systems Interconnection (OSI) end systems, and area addresses. LSPs are used by the receiving routers to maintain their routing tables.
- **IIH** --An IS-IS Hello PDU is used to establish and maintain adjacencies. By default, an Intermediate-to-Intermediate Hello (IIH) is padded to the maximum transmission unit (MTU) size.
- **PSNP** --A partial sequence number PDU (PSNP) contains summaries of only a subset of known LSPs. A PSNP is used to acknowledge and request link-state information by soliciting newer versions of a complete LSP, or acknowledging receipt of an updated LSP, respectively.
- **CSNP** --A complete sequence number PDU (CSNP) contains summaries of all LSPs known by the issuing router.

LSP-Related Intervals and Exponential Backoff Timers

The following timers and intervals relate to LSPs that are generated by the IS-IS router:

- **LSP refresh interval** --Specifies the number of seconds (0 to 65535) the router will wait before refreshing (re-creating and reflooding) its own LSP.
- **Maximum LSP lifetime** --Specifies the value of the lifetime in the LSP header. Lifetime is used by all IS-IS routers in order to age out and purge old LSPs.

The following exponential backoff timers have been implemented in IS-IS to control the events of SPF calculation, Partial Route Calculations (PRC) computation, and LSP generation:

- **PRC interval** --Specifies the number of seconds between two consecutive PRCs. When changes that do not affect the topology, such as advertised external prefixes, are detected, the PRC is triggered.
- **LSP generation interval** --Specifies the number of seconds between creating new versions of a given LSP on a per-node basis.
- **SPF interval** --Specifies the number of seconds between two consecutive SPF calculations.

The purpose of these exponential backoff timers is to react quickly to the first events but, under constant churn, to slow down in order to prevent the CPU of the router from collapsing. The exponential backoff algorithm operates as follows:

1. An initial event triggers the SPF, PRC, or LSP generation.
2. The initial wait time that is configured for the interval determines the time between the initial event and the start of the SPF, PRC, or LSP generation.
3. The incremental wait time that is configured for the interval determines the amount of time that the router will wait in between the consecutive SPF execution, PRC execution, or LSP generation. This incremental value will increase exponentially between the incremental events until the maximum value is reached. For example, the incremental value will be (1x incremental value) between the first and second events, (2 x incremental value) between the second and third event, (4 x incremental value) between the third and fourth event, (8 x incremental value) between the fourth and fifth event, and so on, until the configured maximum interval--amount of time in seconds that the router will wait in between consecutive SPF execution, PRC execution, or LSP generation--has been reached.
4. If no new triggers have been received after two times the configured maximum wait-interval value, the network stabilizes, returning to a steady state and fast behavior. The initial wait-time interval will be reinstated.

See the to configure the recommended settings for the SPF, PRC, and LSP generation timers.

IS-IS Hello PDU Timers

The different IS-IS Hello timers need to be adapted according to the adjacency convergence time required for each subnet. Where a rapid adjacency loss has been detected, the timers need to be reduced. These timers should be modified if necessary after deployment and after an accurate monitoring of the network stability and convergence has occurred.

- **Hello interval** --Number of seconds during two consecutive transmissions of IIH PDUs.
- **Hello interval minimum** --When the hello interval is configured, the hold time is set to one second. The significance of the hello multiplier changes if Fast Hellos are used; the hello multiplier becomes the number of hellos that will be sent per second.
- **Hello multiplier** --An integer from 1 to 300 that is used to calculate the hold time. The hold time is the number of seconds during which the router will wait for an IIH before declaring that its neighbor is lost. The router multiplies the hello interval by the hello multiplier to determine the hold time. To avoid unnecessary adjacency resets, increase the default value of 3 on interfaces where frequent losses of IIH PDUs are detected.
- **IS-IS retransmit interval** --Specifies the number of seconds between the resending of IS-IS link-state PDU transmissions for point-to-point links.

CSNP Interval

TheCSNP interval specifies the number of seconds between the two consecutive transmissions of CSNP PDUs. CSNP are generated by the designated router (DIS) in order for all routers connected to a broadcast media to synchronize their databases and by adjacent routers on a point-to-point network while setting up an adjacency. CSNPs are used to keep all router databases up to date. The lower the value of the CSNP interval, the faster the speed of the sychronization. However, a CSNP interval that is too low will trigger intensive PSNP PDU transmissions. All routers that are not synchronized with the DIS (Designated Intermediate System) and that, therefore, need additional LSPs in their database send PSNPs.

SPF, PRC, and LSP generation exponential backoff timers need to be tuned according to the level of stability of the network and the stability required in the routing domain. For instance, setting low values will trigger a fast convergence with a potential risk of high resource utilization if flapping routes cause network churn. Setting high values will keep the network stable with slower convergence.

It is recommended to leave the default value for the LSP generation interval at 5 seconds and also to increase the maximum lifetime for LSPs to 65,535 seconds, in order to conserve CPU usage for generation and refreshing of LSPs.

If you are using a routing algorithm based on SPF and if you use values for the initial required delay that are fewer than 40 milliseconds, SPF may start before the LSP that triggered SPF is flooded to neighbors. The router should always flood, at least, the LSP that triggered SPF before the router runs the SPF computation. LSP flooding is required in order to guarantee that the network update in the LSP is propagated around the network as quickly as possible.

How to Reduce Link Failure and Topology Change Notification Times in IS-IS Networks

Tuning SPF PRC and LSP Generation Exponential Backoff Timers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*]
5. **prc-interval** *prc-max-wait* [*prc-initial-wait prc-second-wait*]
6. **lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*]
7. **max-lsp-lifetime** [**hours**] *value*
8. **lsp-refresh-interval** *seconds*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Router(config)# router isis`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*] **Example:** `Router(config-router)# spf-interval 5 1 20`	Customizes IS-IS throttling of SPF calculations. **Note** The recommended values for the *spf-max-wait*, *spf-initial-wait*, and *spf-second-wait* arguments are 5, 1, and 20, respectively.
Step 5	**prc-interval** *prc-max-wait* [*prc-initial-wait prc-second-wait*] **Example:** `Router(config-router)# prc-interval 5 1 20`	Customizes IS-IS throttling of PRC calculations. **Note** The recommended values for the *prc-max-wait*, *prc-initial-wait*, and *prc-second-wait* arguments are 5, 1, and 20, respectively.
Step 6	**lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*] **Example:** `Router(config-router)# lsp-gen-interval 5 1 20`	Sets the minimum interval at which LSPs are generated. **Note** The recommended values for the *lsp-max-wait*, *lsp-initial-wait*, and *lsp-second-wait* arguments are 5, 1, and 20, respectively.
Step 7	**max-lsp-lifetime** [**hours**] *value* **Example:** `Router(config-router)# max-lsp-lifetime 65535`	Sets the maximum time for which LSPs persist without being refreshed. • To reduce network resources used for LSP generation, increase the LSP maximum lifetime value of 65535.
Step 8	**lsp-refresh-interval** *seconds* **Example:** `Router(config-router)# lsp-refresh-interval 65000`	Sets the minimum interval at which LSPs are refreshed. • To reduce network resources used for LSP refresh, increase the value to the LSP refresh interval to maximum value of 65000 seconds.
Step 9	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.

Enabling IS-IS Fast Flooding of LSPs

It is recommended that you keep the default values for the **isis retransmit-interval** and **isis retransmit-throttle-interval** commands when you configure the **fast-flood** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **fast-flood** *lsp-number*
5. **end**
6. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Router(config)# router isis`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required, and enters router configuration mode.
Step 4	**fast-flood** *lsp-number* **Example:** `Router(config-router)# fast-flood 20`	Fast-floods LSPs. • It is recommended that you keep the default values for the **isis retransmit-interval** and **isis retransmit-throttle-interval** commands when you configure the **fast-flood** command.
Step 5	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Router# show running-config`	(Optional) Verifies that fast flooding has been enabled.

Monitoring IS-IS Network Convergence Time

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isis display delimiter** [**return** *count* | **character** *count*]
4. **exit**
5. **show isis database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
6. **show isis** [*area-tag*] **routes**
7. **show isis spf-log**
8. **show isis** [*process-tag*] **topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**isis display delimiter** [**return** *count* | **character** *count*] **Example:** `Router(config)# isis display delimiter return 2`	(Optional) Makes output from multiarea displays easier to read by specifying the delimiter to use to separate displays of information.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show isis database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:** `Router# show isis database detail`	(Optional) Displays the IS-IS link-state database.
Step 6	**show isis** [*area-tag*] **routes** **Example:** `Router# show isis financetag routes`	(Optional) Displays the IS-IS Level 1 forwarding table for IS-IS learned routes.

	Command or Action	Purpose
Step 7	**show isis spf-log** **Example:** `Router# show isis spf-log`	(Optional) Displays how often and why the router has run a full SPF calculation.
Step 8	**show isis** [*process-tag*] **topology** **Example:** `Router# show isis financetag topology`	(Optional) Displays a list of all connected routers in all areas. • If a process tag is specified, output is limited to the specified routing process. When "null" is specified for the process tag, output is displayed only for the router process that has no tag specified. If a process tag is not specified, output is displayed for all processes.

Configuration Examples for Reducing Link Failure and Topology Change Notification Times in IS-IS Networks

Example Tuning IS-IS LSP Generation

The following example configures the router to reduce LSP flooding and the consequent resource consumption by tuning the LSP values to their maximums. Adjusting the IS-IS timers will decrease the time required for the router to send routing updates.

```
Router> enable
Router# configure terminal
Router(config)# router isis
Router(config-router)# isis tag 200
Router(config-router)# lsp-gen-interval 5
Router(config-router)# max-lsp-lifetime 65535
Router(config-router)# lsp-refresh-interval 65000
```

Example Tuning IS-IS Fast-Flooding of LSPs

In the following example, the **fast-flood** command is entered to configure the router to flood the first seven LSPs that invoke SPF, before the SPF computation is started. When the **show running-config** command is entered, the output confirms that fast-flooding has been enabled on the router.

```
Router> enable
Router# configure terminal
Router(config)# router isis first
Router(config-router)# fast-flood 7
Router(config-router)# end
Router# show running-config | include fast-flood

fast-flood 7
```

Where to Go Next

To configure features to improve IS-IS network convergence times and scalability, complete the optional tasks in one or more of the following modules:

- "Setting Best Practice Parameters for IS-IS Fast Convergence"
- "Reducing Failure Detection Times in IS-IS Networks"
- "Reducing Alternate-Path Calculation Times in IS-IS Networks"

Additional References

The following sections provide references related to IS-IS configuration tasks to achieve fast convergence and scalability.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of Cisco IS-IS conceptual information with links to all the individual IS-IS modules	"Integrated IS-IS Routing Protocol Overview"
Customizing IS-IS for fast convergence and scalability	"Overview of IS-IS Fast Convergence" module
IPv6 Routing: IS-IS Multitopology Support for IPv6	"*IPv6 Routing: IS-IS Multitopology Support for IPv6* " module

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Reducing Link Failure and Topology Change Notification Times in IS-IS Networks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 169

Enabling Enhanced IS-IS Fast Flooding of LSPs

Overview

This chapter introduces the Enhanced IS-IS Fast Flooding feature. Designed to increase the rate of LSP flooding, it enables quicker network adaptation to topology changes and reduces convergence times.

Unlike the standard LSP flooding mechanisms, this feature is not enabled by default and offers a configurable environment to efficiently manage LSP transmission rates.

Restrictions

- The feature must be manually enabled as it is not active by default.

Information About Enabling Enhanced IS-IS Fast Flooding of LSPs

Enabling Enhanced IS-IS Fast Flooding of LSPs

The IS-IS LSP Fast Flooding feature is designed to optimize the transmission of Link State Packets (LSPs) within an IS-IS domain. By increasing the flooding rate of LSPs, this feature allows for faster dissemination of topology information, facilitating quicker network convergence in response to changes.

Administrators can configure the feature to send LSPs in bursts, targeting a default rate of 1000 LSPs per second, significantly higher than the rate achieved with standard settings. The feature dynamically adjusts the

sending rate based on the neighbor's ability to process and acknowledge the LSPs, ensuring efficient communication and preventing overloading.

While standard IS-IS flooding operates on a fixed timer, the Fast Flooding feature introduces flexibility and control, with several configurable parameters available through CLI commands. This includes settings for the local PSNP interval and the maximum flooding rate, which can be adjusted according to network requirements.

The Fast Flooding feature is inactive by default, providing network operators the choice to enable it selectively where the network infrastructure will benefit from enhanced LSP flooding capabilities. This selective activation is crucial for maintaining optimal performance across various network topologies and conditions.

Adaptive Flooding Rate Adjustment

The Enhanced IS-IS Fast Flooding feature includes an intelligent mechanism for adapting the LSP flooding rate in real-time. This dynamic adjustment is based on continuous monitoring of the acknowledgment rates (PSNP receipt) from neighboring routers. When the feature detects delays in acknowledgment, it automatically reduces the flooding rate to prevent overloading the neighbor's processing capabilities. Conversely, if acknowledgments are received promptly, the feature may increase the flooding rate up to the configured maximum, optimizing the speed of topology dissemination.

This adaptive approach ensures that the feature responds appropriately to the operational conditions of the network, providing an optimal balance between fast convergence and network stability. Network administrators can use this information to fine-tune the feature's parameters, ensuring that the flooding rate is both efficient and sustainable.

How to Configure Enhanced IS-IS Fast Flooding of LSPs

Configure the Enhanced IS-IS Fast Flooding feature using the following CLI commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis 1 lsp-fastflooding**
4. **router isis 1 lsp-fastflooding max-lsp-tx 2000**
5. **router isis 1 lsp-fastflooding remotepsnp-delay 500**
6. **router isis 1 psnpinterval 2000**
7. **interface GigabitEthernet 0/0/0 isis remote-psnpdelay 400**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**router isis 1 lsp-fastflooding**
	Example:
	`Device# configure terminal`
	`Device(config)# router isis 1`
	`Device(configrouter)# lspfast-flooding`
		Activates the fast flooding feature for the IS-IS process.

Step 4	**router isis 1 lsp-fastflooding max-lsp-tx 2000** **Example:** `Device(configrouter)# lspfast-flooding max-lsp-tx 2000`	(Optional) Configures the maximum rate at which LSPs are sent, in this case, 2000 LSPs per second.
Step 5	**router isis 1 lsp-fastflooding remotepsnp-delay 500** **Example:** `Device(configrouter)# lspfast-flooding remote-psnp-delay 500`	(Optional) Specifies the delay, in milliseconds, that the router expects a PSNP acknowledgment from neighbors after sending an LSP. Here, it is set to 500 milliseconds.
Step 6	**router isis 1 psnpinterval 2000** **Example:** `Device(configrouter)# psnpinterval 2000`	(Optional) Sets the interval at which PSNPs are sent. This configuration increases the interval to 2000 milliseconds.

Step 7	**interface GigabitEthernet 0/0/0 isis remote-psnpdelay 400**
	Example:
	`Device(config)# interface GigabitEthernet0/0/0`
	`Device(configif)# isis remotepsnp-delay 400`
		(Optional) Overrides the default or global remote PSNP delay for a specific interface, such as GigabitEthernet0/0/0, with a custom value of 400 milliseconds.

Step 8	**end** **Example:** `Router(configrouter)# end`	Returns to privileged EXEC mode.

Configuration Examples for Enabling Enhanced IS-IS Fast Flooding

The examples provided above demonstrate how to enable the Enhanced ISIS Fast Flooding feature and customise further options to manage the feature effectively.

To enable the IS-IS Fast Flooding feature on your device, enter the following commands:

```
Device# configure terminal
Device(config)# router isis 1
Device(config-router)# lsp-fast-flooding
Device(config-isis-fspeed)#?
ISIS flood speed configuration commands:
default                     Set a command to its defaults
exit-lsp-fast-flooding      Exit from LSP Fast Flooding mode
max-lsp-tx                  Maximum LSP Transmit Rate in LSP/Sec
no                          Negate a command or set its defaults
remote-psnp-delay           Remote PSNP delay
Device(config-isis-fspeed)#
```

Configure the router to send LSPs at a maximum rate of 2000 LSPs per second by using the following commands:

```
Device# configure terminal
Device(config)# router isis 1
Device(config-router)# lsp-fast-flooding
Device(config-router)# lsp-fast-flooding max-lsp-tx 2000
Device(config-router)# exit
```

Specify the PSNP acknowledgment delay to 500 milliseconds with these commands:

```
Device# configure terminal
Device(config)# router isis 1
Device(config-router)# lsp-fast-flooding
Device(config-router)# lsp-fast-flooding remote-psnp-delay 500
Device(config-router)# exit
```

Set the PSNP interval timer to 2000 milliseconds as shown below:

```
Device# configure terminal
Device(config)#router isis 1
Device(config-router)#lsp-fast-flooding
Device(config-isis-fspeed)#max-lsp-tx 2000
Device(config-isis-fspeed)#exit
Device#
```

Override the global PSNP delay with an interface-specific value on GigabitEthernet0/0/0:

```
Device# configure terminal
Device(config)# interface GigabitEthernet0/0/0
Device(config-if)# isis remote-psnp-delay 400
Device(config-if)# exit
```

Feature Information for Enabling Enhanced IS-IS Fast Flooding of LSPs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Table 190: Table 1. Feature Information for Enabling Enhanced IS-IS Fast Flooding of LSPs

Feature Name	Releases	Feature Information
Enabling Enhanced IS-IS Fast Flooding of LSPs	Cisco IOS XE 17.14.1a	The Enhanced IS-IS Fast Flooding of LSPs feature accelerates network convergence by improving the speed at which Link State Packets (LSPs) are spread across an IS-IS network. This is accomplished by adjusting the LSP transmission rate to match the receiving routers' capabilities, ensuring efficient and rapid dissemination of network topology information. Implemented from Cisco IOS XE 17.14.1a onwards, this feature is configured with the **router isis lsp-fast-flooding** command. For more precise control, options such as **max-lsp-tx**, **psnp-interval**, and **per-interface** settings are available within the router isis configuration. The isis remote-psnp-delay command further enhances LSP flooding customization. By default, this feature is off and requires manual activation. Once enabled, it provides quicker network response to changes, promoting high network availability and robustness.

CHAPTER 170

IS-IS Support for Route Tags

The IS-IS Support for Route Tags feature enables you to tag Intermediate System-to-Intermediate System (IS-IS) route prefixes and use those tags in a route map to control IS-IS route redistribution or route leaking. The results are network scalability and faster convergence for device updates.

Prerequisites for IS-IS Support for Route Tags

Because the Intermediate System-to-Intermediate System (IS-IS) route tag will be used in a route map, you must understand how to configure a route map.

To use the route tag, you must configure the **metric-style wide** command. (The **metric-style narrow** command is configured by default.) The tag value is set into sub-TLV 1 for type, length, values (TLV) Type 135.

You must understand the task for which you are using the route tag, such as route redistribution, route summarization, or route leaking.

You should be familiar with the concepts described in the “Overview of IS-IS Fast Convergence” module.

Before you tag any IS-IS routes, you need to make the following decisions:

- Your goal to set values for routes or redistribute routes (or both).
- Where in your network you want to tag routes.
- Where in your network you want to reference the tags.
- Which tagging method you will use. This method determines which task to perform.

Information About IS-IS Support for Route Tags

Route Redistribution

Devices are allowed to redistribute external prefixes, or routes, that are learned from any other routing protocol, static configuration, or connected interfaces. The redistributed routes are allowed in either a Level 1 device or a Level 2 device. Level 2 routes injected as Level 1 routes is called route leaking.

IS-IS Caching of Redistributed Routes

Intermediate System-to-Intermediate System (IS-IS) caches routes that are redistributed from other routing protocols or from another IS-IS level into a local redistribution cache that is maintained by IS-IS. Caching occurs automatically and requires no configuration. The caching of redistributed routes improves IS-IS convergence time when routes are being redistributed into IS-IS. IS-IS caching of redistributed routes increases the performance of link-state packet (LSP) protocol data unit (PDU) generation, significantly improving network scalability.

Prioritize the Update of IP Prefixes in the RIB to Reduce Alternate-Path Calculation Time

The time needed for the IS-IS Routing Information Base (RIB) or routing table to update depends on the number of changed Intermediate System-to-Intermediate System (IS-IS) prefixes or routes that must be updated. You can tag important IS-IS IP prefixes and configure the device to give priority to the tagged prefixes so that high-priority prefixes are updated first in the RIB. For example, the loopback addresses for the devices in a Multiprotocol Label Switching (MPLS) VPN environment are considered high-priority prefixes.

IS-IS Priority-Driven IP Prefix RIB Installation

In a network where devices run the Intermediate System-to-Intermediate System (IS-IS) protocol, convergence is achieved when a consistent view of the topology is distributed to all devices in the network. When a network event causes a topology change, a number of steps must occur in order for convergence to occur. The device that initially detects the topology change (for example, an interface state change) must inform other devices of the topology change by flooding updated routing information (in the form of link-state protocol data units [PDUs]) to other devices. All devices, including the device that detected the topology change, must utilize the updated topology information to recompute shortest paths (run a shortest path first [SPF]), providing the updated output of the SPF calculation to the device's routing information base (RIB), which eventually causes the updated routing information to be used to forward packets. Until all devices have performed these basic steps, some destinations might be temporarily unreachable. Faster convergence benefits the network performance by minimizing the period of time during which stale topology information—the previous routing information that will be obsoleted by the updated routing information—is used to forward packets.

After performing an SPF, IS-IS must install updated routes in the RIB. If the number of prefixes advertised by IS-IS is large, the time between the installation of the first prefix and the last prefix is significant. Priority-driven IP prefix RIB installation allows a subset of the prefixes advertised by IS-IS to be designated as having a higher priority. Updates to the paths to these prefixes are installed before updates to prefixes that do not have this designation. Priority-driven IP prefixes reduce the convergence time for the important IS-IS

IP prefixes and results in faster updating for routes that are dependent on these prefixes. Faster updates shortens the time during which stale information is used for forwarding packets to these destinations.

Prefixes are characterized as having one of three levels of importance:

1. High-priority prefixes—prefixes that are tagged with a tag designated for fast convergence.
2. Medium-priority prefixes—any /32 prefixes that are not designated as high-priority prefixes.
3. Low-priority prefixes—all other prefixes.

When IS-IS updates the RIB, prefixes are updated in the order based on the associated level of importance.

When you assign a high-priority tag to some IS-IS IP prefixes, those prefixes with the higher priority are updated in the routing tables before prefixes with lower priority. In some networks, the high-priority prefixes are the provider edge (PE) loopback addresses. The convergence time is reduced for the important IS-IS IP prefixes and results in reduced convergence time for the update processes that occur in the global RIB and Cisco Express Forwarding.

IS-IS Routes Tagged to Control Their Redistribution

You can control the redistribution of Intermediate System-to-Intermediate System (IS-IS) routes by tagging them. The term “route leaking” refers to controlling distribution through tagging of routes.

How Route Summarization Can Enhance Scalability in IS-IS Networks

Summarization is a key factor that enhances the scalability of a routing protocol. Summarization reduces the number of routing updates that are flooded across areas or routing domains. For example, in multiarea Intermediate System-to-Intermediate System (IS-IS) networks, a good addressing scheme can optimize summarization by not allowing an overly large Level 2 database to be unnecessarily populated with updates that have come from Level 1 areas.

A device can summarize prefixes on redistribution whether the prefixes have come from internal prefixes, local redistribution, or Level 1 device redistribution. Routes that have been leaked from Level 2 to Level 1 and routes that are advertised into Level 2 from Level 1 can also be summarized.

Benefits of IS-IS Route Tags

The IS-IS Support for Route Tags feature allows you to tag IP addresses of an interface and use the tag to apply administrative policy with a route map.

You can tag Intermediate System-to-Intermediate System (IS-IS) routes to control their redistribution. You can configure a route map to set a tag for an IS-IS IP prefix (route) or match on the tag (perhaps on a different device) to redistribute IS-IS routes. Although the **match tag** and **set tag** commands existed for other protocols before the IS-IS Support for Route Tags feature, they were not implemented for IS-IS, so they did nothing when specified in an IS-IS network.

You can tag a summary route and then use a route map to match the tag and set one or more attributes for the route.

IS-IS Route Tag Characteristics

An Intermediate System-to-Intermediate System (IS-IS) route tag number can be up to 4 bytes long. The tag value is set into a sub-TLV 1 for type, length, values (TLV) Type 135.

Only one tag can be set to an IS-IS IP route (prefix). The tag is sent in link-state packet (LSP) protocol data units (PDUs) advertising the route. Setting a tag to a route alone does nothing for your network. You can use the route tag at area or Level 1/Level 2 boundaries by matching on the tag and then applying administrative policies such as redistribution, route summarization, or route leaking.

Configuring a tag for an interface (with the **isis tag** command) triggers the generation of new LSPs from the device because the tag is new information for the PDUs.

IS-IS Route Leaking Based on a Route Tag

You can tag Intermediate System-to-Intermediate System (IS-IS) routes to configure route leaking (redistribution). Because only the appropriate routes are redistributed—or leaked—the results is network scalability and faster convergence for the device update. If you configure route leaking and you want to match on a tag, use a route map (not a distribute list).

There are two general steps to using IS-IS route tags: tagging routes and referencing the tag to set values for the routes or redistribute routes.

There are three ways to tag IS-IS routes: tag routes for networks directly connected to an interface, set a tag in a route map, or tag a summary route. The tagging method is independent of how you use the tag.

After you tag the routes, you can use the tag to set values (such as metric, next hop, and so on) or redistribute routes. You might tag routes on one device, but reference the tag on other devices, depending on what you want to achieve. For example, you could tag the interface on Device A with a tag, match the tag on Device B to set values, and redistribute routes on Device C based on values using a route map.

Limit the Number of Routes That Are Redistributed into IS-IS

If you mistakenly inject a large number of IP routes into an Intermediate System-to-Intermediate System (IS-IS), perhaps by redistributing Border Gateway Protocol (BGP) into IS-IS, the network can be severely flooded. You can limit the number of redistributed routes prevents this potential problem. You can either configure IS-IS to stop allowing routes to be redistributed once your maximum configured value is reached or configure the software to generate a system warning once the number of redistributed prefixes reaches the maximum value.

In some cases when a limit is not placed on the number of redistributed routes, the link-state packet (LSP) might become full and routes might be dropped. You can specify which routes should be suppressed in that event so that the consequence of an LSP full state is handled in a graceful and predictable manner.

Redistribution is usually the cause of the LSP full state. By default, external routes redistributed into IS-IS are suppressed if the LSP full state occurs. IS-IS can have 255 fragments for an LSP in a level. When no space is left in any of the fragments, an LSPFULL error message is generated.

Once the problem that caused the LSP full state is resolved, you can clear the LSPFULL state.

Note You cannot both limit redistributed prefixes and also choose to be warned only.

Streamline the Routing Table Update Process by Excluding Connected IP Prefixes from LSP Advertisements

To speed up Intermediate System-to-Intermediate System (IS-IS) convergence time, limit the number of IP prefixes carried in link-state packets (LSPs). Configuring interfaces as unnumbered will limit the prefixes. However, for network management reasons, you might want to have numbered interfaces and also want to prevent advertising interface addresses into IS-IS. Two alternative methods avoid the overpopulation of routing tables and thereby reduce IS-IS convergence time. To choose the method that works best for your network type, you should become familiar with the concepts described in the following sections:

Small-Scale Method to Reduce IS-IS Convergence Time

You can explicitly configure an Intermediate System-to-Intermediate System (IS-IS) interface not to advertise its IP network to the neighbors (by using the **no isis advertise-prefix** command). This method is feasible for a small network; it does not scale well. If you have dozens or hundreds of devices in your network, with possibly ten times as many physical interfaces involved, adding this command to each device's configuration is not practical.

Large-Scale Method to Reduce IS-IS Convergence Time

A way to reduce Intermediate System-to-Intermediate System (IS-IS) convergence is to configure the IS-IS instance on a device to advertise only passive interfaces (by using the **advertise-passive-only** command). This command relies on the fact that a user enabling IS-IS on a loopback interface usually configures the loopback as passive (to prevent sending unnecessary hello PDUs through it because there is no chance of finding a neighbor behind it). Thus, if you want to advertise only the loopback and if it has already been configured as passive, configuring the **advertise-passive-only** command per IS-IS instance would prevent the overpopulation of the routing tables.

Benefit of Excluding IP Prefixes of Connected Networks in LSP Advertisements

Whether you choose to prevent the advertising of Intermediate System-to-Intermediate System (IS-IS) interface subnetworks or to advertise only the IS-IS prefixes that belong to passive (loopback) interfaces, you will reduce IS-IS convergence time. The IS-IS Mechanisms to Exclude Connected IP Prefixes from LSP Advertisements feature is recommended in any case where fast convergence is required.

How to Configure IS-IS Support for Route Tags

Configuring IS-IS Incremental SPF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **ispf** [**level-1** | **level-2** | **level-1-2**] [*seconds*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Enables Intermediate System-to-Intermediate System (IS-IS) as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**ispf** [**level-1** | **level-2** | **level-1-2**] [*seconds*] **Example:** `Device(config-router)# ispf level-1-2 60`	Enables IS-IS incremental SPF. • The *seconds* argument represents the number of seconds after configuring this command that incremental SPF is activated. The range is 1 to 600. The default value is 120 seconds. The *seconds* argument applies only when you have enabled IS-IS.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Assigning a High Priority Tag to an IS-IS IP Prefix

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip router isis** [*area-tag*]
5. **isis tag** *tag-value*
6. **exit**
7. **router isis** [*area-tag*]
8. **ip route priority high tag** *tag-value*
9. **end**
10. **show isis rib** [*ip-address* | *ip-address-mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip router isis** [*area-tag*] **Example:** `Router(config-if)# ip router isis tag13`	Enables IS-IS as an IP routing protocol, and assigns a tag to a process, if required. **Note** If the *area-tag* argument is not specified, a null tag is assumed and the process is referenced with a null tag. This name must be unique among all IP or Connectionless Network Service (CLNS) router processes for a given router.
Step 5	**isis tag** *tag-value* **Example:** `Router(config-if)# isis tag 17`	Sets a tag on the IP address configured for an interface when this IP prefix is put into an IS-IS LSP. • The *tag-value* argument requires an interger in a range from 1 to 4294967295 and serves as a tag on an IS-IS route.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 7	**router isis** [*area-tag*] **Example:** `Router(config)# router isis marketing`	Enables the IS-IS routing protocol and specifies an IS-IS process. Enters router configuration mode. **Note** If the *area-tag* argument is not specified, a null tag is assumed and the process is referenced with a null tag. This name must be unique among all IP or CLNS router processes for a given router.
Step 8	**ip route priority high tag** *tag-value* **Example:**	Assigns a high priority to prefixes associated with the specified tag value.

	Command or Action	Purpose
	`Router(config-router)# ip route priority high tag 17`	• Assigns a high priority to IS-IS IP prefixes with a specific route tag in a range from 1 to 4294967295 that you specify for the *tag-value* argument.
Step 9	**end** **Example:** `Router(config-router)# end`	(Optional) Saves configuration commands to the running configuration file and returns to privileged EXEC mode.
Step 10	**show isis rib** [*ip-address* | *ip-address-mask*] **Example:** `Router# show isis rib 255.255.255.0`	Displays paths for a specific route in the IP Version 4 IS-IS local RIB. • IS-IS maintains a local database for all IS-IS routing information. This local database is referred to as the IS-IS local RIB. It contains additional attributes that are not maintained in the global IP routing table. Access to the contents of the local RIB is used to support the **show isis rib** command, which is used here to verify routing information related to the Priority-Driven IP Prefix RIB Installation feature.

Troubleshooting Tips

You can enter the **debug isis rib local** command to verify whether the IP prefixes that are advertised by Intermediate System-to-Intermediate System (IS-IS) link-state packet (LSP) protocol data units (PDUs) are being updated correctly in the IS-IS local Routing Information Base (RIB).

Tagging Routes for Networks Directly Connected to an Interface

Before you begin

- Because the IS-IS route tag will be used in a route map, you must understand how to configure a route map.
- In order to use the route tag, you must configure the **metric-style wide command. (The metric-style narrow**command is configured by default**). The tag value is set into sub-TLV 1 for TLV (Type Length Value) Type 135.**
- You must understand the task for which you are using the route tag, such as route redistribution, route summarization, or route leaking.

Before you tag any IS-IS routes, you need to decide on the following:

1. Your goal to set values for routes or redistribute routes (or both).
2. Where in your network you want to tag routes.
3. Where in your network you want to reference the tags.
4. Which tagging method you will use, which determines which task in this section to perform.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip address** *ip-address mask* **secondary**
6. **isis tag** *tag-value*
7. **end**
8. **show isis database verbose**
9. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Configures an interface.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for an interface. • In this example, the network 10.1.1.0 will be tagged.
Step 5	**ip address** *ip-address mask* **secondary** **Example:** `Router(config-if)# ip address 10.2.2.1 255.255.255.0 secondary`	(Optional) Sets a secondary IP address for an interface. • In this example, the network 10.2.2.0 will be tagged.
Step 6	**isis tag** *tag-value* **Example:** `Router(config-if)# isis tag 120`	Sets a tag on the IP addresses configured under this interface when those IP prefixes are put into an IS-IS LSP. • The tag must be an integer.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Exits configuration mode and returns to privileged EXEC mode.
Step 8	**show isis database verbose** **Example:** `Router# show isis database verbose`	(Optional) Displays details about the IS-IS link-state database, including the route tag. • Perform this step if you want to verify the tag.
Step 9	**show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*]] **Example:** `Router# show ip route 10.1.1.1 255.255.255.0`	(Optional) Displays the current state of the routing table. • Perform this step if you want to verify the tag.

What to Do Next

Applying the tag does nothing of value for your network until you use the tag by referencing it in a route map, either to set values, to redistribute routes, or to do both. Proceed to the section "Using the Tag to Set Values or Redistribute Routes."

Tagging Routes Using a Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match tag** *tag-value* [*...tag-value*]
5. Use an additional **match** command for each match criterion that you want.
6. **set tag** *tag-value*
7. Set another value, depending on what else you want to do with the tagged routes.
8. Repeat Step 7 for each value that you want to set.
9. Repeat Steps 3 through 8 for each route-map statement that you want.
10. **end**
11. **show isis database verbose**
12. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | [**list** *access-list-number* | [*access-list-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map static-color permit 15`	Defines the conditions for redistributing routes from one routing protocol into another or from one IS-IS level to another. • This command causes the router to enter route-map configuration mode.
Step 4	**match tag** *tag-value* [*...tag-value*] **Example:** `Router(config-route-map)# match tag 15`	(Optional) Matches routes tagged with the specified tag numbers. • If you are setting a tag for the first time, you cannot match on tag; this step is an option if you are changing tags.
Step 5	Use an additional **match** command for each match criterion that you want.	(Optional) See the appropriate **match** commands in the *Cisco IOS IP Routing: Protocol-Independent Command Reference* • Repeat this step for each match criterion you that want.
Step 6	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 10`	Specifies the tag number to set.
Step 7	Set another value, depending on what else you want to do with the tagged routes.	(Optional) See the following **set** commands in the *Cisco IOS IP Routing: Protocol-Independent Command Reference* • **set level** • **set metric** • **set metric-type**
Step 8	Repeat Step 7 for each value that you want to set.	(Optional)
Step 9	Repeat Steps 3 through 8 for each route-map statement that you want.	(Optional)

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-route-map)# end`	(Optional) Exits configuration mode and returns to privileged EXEC mode.
Step 11	**show isis database verbose** **Example:** `Router# show isis database verbose`	(Optional) Displays details about the IS-IS link-state database, including the route tag. • Perform this step if you want to verify the tag.
Step 12	**show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | [**list** *access-list-number* | [*access-list-name*]] **Example:** `Router# show ip route 10.1.1.1 255.255.255.0`	(Optional) Displays the current state of the routing table. • Perform this step if you want to verify the tag.

What to Do Next

Applying the tag does nothing of value for your network until you use the tag by referencing it in a route map, either to set values, to redistribute routes, or to do both. Proceed to the section "Using the Tag to Set Values and or Redistribute Routes."

Tagging a Summary Address

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis [** *area-tag* **]**
4. **metric-style wide**
5. **summary-address** *address mask* {**level-1** | **level-1-2** | **level-2**} [**tag** *tag-value*] [**metric** *metric-value*]
6. **end**
7. **show isis database verbose**
8. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | [**list** *access-list-number* | [*access-list-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis [** *area-tag* **]** **Example:** `Router(config)# router isis`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router running IS-IS so that it generates and accepts type, length, and value object (TLV) 135 for IP addresses.
Step 5	**summary-address** *address mask* {**level-1** | **level-1-2** | **level-2**} [**tag** *tag-value*] [**metric** *metric-value*] **Example:** `Router(config-router)# summary-address 192.168.0.0 255.255.0.0 tag 12345 metric 321`	Creates aggregate addresses for IS-IS. **Note** If a tagged route is summarized and the tag is not explicitly configured in the **summary-address** command, then the tag is lost.
Step 6	**end** **Example:** `Router(config-router)# end`	(Optional) Exits configuration mode and returns to privileged EXEC mode.
Step 7	**show isis database verbose** **Example:** `Router# show isis database verbose`	(Optional) Displays details about the IS-IS link-state database, including the route tag. • Perform this step if you want to verify the tag.
Step 8	**show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | [**list** *access-list-number* | [*access-list-name*]] **Example:** `Router# show ip route 10.1.1.1 255.255.255.0`	(Optional) Displays the current state of the routing table. • Perform this step if you want to verify the tag.

What to Do Next

Applying the tag does nothing of value for your network until you use the tag by referencing it in a route map to set values. It is unlikely that you will redistribute summary routes. Proceed to the “Using the Tag to Set Values or Redistribute Routes” section.

Using the Tag to Set Values and or Redistribute Routes

Before you begin

You must have already applied a tag on the interface, in a route map, or on a summary route. See the IS-IS Routes Tagged to Control Their Redistribution, on page 2165.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match tag** *tag-value*
5. Specify a **match** command for each match criterion that you want.
6. Set a value, depending on what you want to do with the tagged routes.
7. Repeat Step 6 for each value that you want to set.
8. Repeat Steps 3 through 7 for each route-map statement that you want.
9. **exit**
10. **router isis**
11. **metric-style wide**
12. **redistribute** *protocol* [*process-id*] [**level-1**| **level-1-2** | **level-2**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] **Example:** `Router(config)# route-map static-color permit 15`	Defines the conditions for redistributing routes from one routing protocol into another or from one IS-IS level to another. • This command causes you to enter route-map configuration mode.
Step 4	**match tag** *tag-value* **Example:** `Router(config-route-map)# match tag 120`	(Optional) Applies the subsequent **set** commands to routes that match routes tagged with this tag number.

	Command or Action	Purpose
Step 5	Specify a **match** command for each match criterion that you want.	(Optional) Reference the appropriate **match** commands in the *Cisco IOS IP Routing: Protocol-Independent Command Reference*.
Step 6	Set a value, depending on what you want to do with the tagged routes.	(Optional) See the following **set** commands in the *Cisco IOS IP Routing: Protocol-Independent Command Reference*. • **set level** • **set metric** • **set metric-type**
Step 7	Repeat Step 6 for each value that you want to set.	(Optional)
Step 8	Repeat Steps 3 through 7 for each route-map statement that you want.	(Optional)
Step 9	**exit** **Example:** `Router(config-route-map)# exit`	(Optional) Returns to global configuration mode.
Step 10	**router isis** **Example:** `Router(config)# router isis`	(Optional) Enables the IS-IS routing protocol and specifies an IS-IS process.
Step 11	**metric-style wide** **Example:** `Router(config-router)# metric-style wide`	Configures a router running IS-IS so that it generates and accepts type, length, and value object (TLV) 135 for IP addresses.
Step 12	**redistribute** *protocol* [*process-id*] [**level-1**| **level-1-2** | **level-2**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-tag*] **Example:** `Router(config-router)# redistribute static ip metric 2 route-map static-color`	(Optional) Redistributes routes from one routing domain into another routing domain.

Limiting the Number of IS-IS Redistributed Routes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]

4. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*]
5. **redistribute maximum-prefix** *maximum* [*percentage*] [**warning-only** | **withdraw**]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Enables Intermediate System-to-Intermediate System (IS-IS) as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] **Example:** `Device(config-router)# redistribute eigrp 10 level-1`	Redistributes routes from one routing domain into another routing domain.
Step 5	**redistribute maximum-prefix** *maximum* [*percentage*] [**warning-only** | **withdraw**] **Example:** `Device(config-router)# redistribute maximum-prefix 1000 80`	Sets a maximum number of IP prefixes that are allowed to be redistributed into IS-IS. • There is no default value for the *maximum* argument. • The *percentage* value defaults to 75 percent. • If the **withdraw** keyword is specified and the maximum number of prefixes is exceeded, IS-IS rebuilds the link-state protocol data unit (PDU) fragments without the external IP prefixes. That is, the redistributed prefixes are removed from the PDUs. **Note** If the **warning-only** keyword had been configured in this command, no limit would be enforced; a warning message would be logged.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode.

Requesting a Warning About the Number of Prefixes Redistributed into IS-IS

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] **match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*]
5. **redistribute maximum-prefix** *maximum* [*percentage*] [**warning-only** | **withdraw**]
6. **lsp-full suppress** {[**external**] [**interlevel**] | **none**}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Enables Intermediate System-to-Intermediate System (IS-IS) as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] **match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] **Example:** `Device(config-router)# redistribute eigrp 10 level-1`	Redistributes routes from one routing domain into another routing domain.

	Command or Action	Purpose
Step 5	**redistribute maximum-prefix** *maximum* [*percentage*] [**warning-only** | **withdraw**] **Example:** `Device(config-router)# redistribute maximum-prefix 1000 80 warning-only`	Causes a warning message to be logged when the maximum number of IP prefixes are redistributed into IS-IS. • Because the **warning-only** keyword is included, no limit is imposed on the number of redistributed prefixes into IS-IS. • There is no default value for the *maximum* argument. • The *percentage* value defaults to 75 percent. • In this example configuration, two warnings are generated: one at 80 percent of 1000 (800 prefixes redistributed) and another at 1000 prefixes redistributed.
Step 6	**lsp-full suppress** {[**external**] [**interlevel**] | **none**} **Example:** `Device(config-router)# lsp-full suppress external interlevel`	(Optional) Controls which routes are suppressed when the link-state packet (LSP) protocol data unit (PDU) becomes full. • The default is **external** (redistributed routes are suppressed). • The **interlevel** keyword causes routes from another level to be suppressed. • The **external** and **interval** keywords can be specified together or separately.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode.

Excluding Connected IP Prefixes on a Small Scale

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address netmask*
5. **no ip directed-broadcast**
6. **ip router isis [area-** *tag* **]**
7. **no isis advertise-prefix**
8. **exit**
9. Repeat Steps 3 through 8 for each interface on which you do not want to advertise IP prefixes.
10. **router isis [area-** *tag* **]**

11. **net** *network-entity-title*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address netmask* **Example:** `Router(config-if)# ip address 192.168.20.1 255.255.255.0`	Sets a primary IP address for an interface. • The network mask can be indicated as a 4-part dotted decimal address or as a prefix. This example uses a 4-part dotted decimal number.
Step 5	**no ip directed-broadcast** **Example:** `Router(config-if)# no ip directed-broadcast`	(Optional) Disables the translation of a directed broadcast to physical broadcasts.
Step 6	**ip router isis [area-** *tag* **]** **Example:** `Router(config-if)# ip router isis`	Configures an IS-IS routing process for IP on an interface and attaches an area designator to the routing process.
Step 7	**no isis advertise-prefix** **Example:** `Router(config-if)# no isis advertise-prefix`	Prevents the advertising of IP prefixes of connected networks in LSP advertisements per IS-IS interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 9	Repeat Steps 3 through 8 for each interface on which you do not want to advertise IP prefixes.	(Optional)

	Command or Action	Purpose
Step 10	**router isis [area-** *tag* **]** **Example:** `Router(config)# router isis`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 11	**net** *network-entity-title* **Example:** `Router(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00`	Configures an IS-IS network entity title (NET) for the routing process.
Step 12	**end** **Example:** `Router(config-router)# end`	(Optional) Saves configuration commands to the running configuration file, exits configuration mode, and returns to privileged EXEC mode.

Excluding Connected IP Prefixes on a Large Scale

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *number*
4. **ip address** *ip-address netmask*
5. **no ip directed-broadcast**
6. **exit**
7. **interface** *type number*
8. **ip address** *ip-address netmask*
9. **no ip directed-broadcast**
10. **ip router isis [area-** *tag* **]**
11. **exit**
12. **router isis [area-** *tag* **]**
13. **passive-interface** [**default**] *type number*
14. **net** *network-entity-title*
15. **advertise-passive-only**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *number* **Example:** `Router(config)# interface loopback 0`	Configures a loopback interface and enters interface configuration mode.
Step 4	**ip address** *ip-address netmask* **Example:** `Router(config-if)# ip address 192.168.10.1 255.255.255.255`	Sets a primary IP address for an interface. • The network mask can be indicated as a 4-part dotted decimal address or as a prefix. This example uses a 4-part dotted decimal number.
Step 5	**no ip directed-broadcast** **Example:** `Router(config-if)# no ip directed-broadcast`	(Optional) Disables the translation of a directed broadcast to physical broadcasts.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0`	Configures an interface type and enters interface configuration mode.
Step 8	**ip address** *ip-address netmask* **Example:** `Router(config-if)# ip address 192.168.20.1 255.255.255.0`	Sets a primary IP address for an interface. • The network mask can be indicated as a 4-part dotted decimal address or as a prefix. This example uses a 4-part dotted decimal number.
Step 9	**no ip directed-broadcast** **Example:** `Router(config-if)# no ip directed-broadcast`	(Optional) Disables the translation of a directed broadcast to physical broadcasts.
Step 10	**ip router isis [area-** *tag* **]** **Example:** `Router(config-if)# ip router isis`	Configures an IS-IS routing process for IP on an interface and attaches an area designator to the routing process.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 12	**router isis [area-** *tag* **]** **Example:** `Router(config)# router isis`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 13	**passive-interface** [**default**] *type number* **Example:** `Router(config-router)# passive-interface loopback 0`	Disables sending routing updates on an interface.
Step 14	**net** *network-entity-title* **Example:** `Router(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00`	Configures an IS-IS NET for the routing process.
Step 15	**advertise-passive-only** **Example:** `Router(config-router)# advertise-passive-only`	Configures IS-IS to advertise only prefixes that belong to passive interfaces.
Step 16	**end** **Example:** `Router(config-router)# end`	(Optional) Saves configuration commands to the running configuration file, exits configuration mode, and returns to privileged EXEC mode.

Monitoring IS-IS Network Convergence Time

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isis display delimiter** [**return** *count* | **character** *count*]
4. **exit**
5. **show isis database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**]
6. **show isis** [*area-tag*] **route**
7. **show isis** [*area-tag*] [**ipv6** | *****] **spf-log**
8. **show isis** [*process-tag*] **topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**isis display delimiter** [**return** *count* | **character** *count*] **Example:** `Device(config)# isis display delimiter return 2`	Makes output from multiarea displays easier to read by specifying the delimiter to use to separate displays of information.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show isis database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [**lspid**] **Example:** `Device# show isis database detail`	Displays the Intermediate System-to-Intermediate System (IS-IS) link-state database.
Step 6	**show isis** [*area-tag*] **route** **Example:** `Device# show isis financetag route`	Displays the IS-IS Level 1 forwarding table for IS-IS learned routes.
Step 7	**show isis** [*area-tag*] [**ipv6** | *****] **spf-log** **Example:** `Device# show isis spf-log`	Displays how often and why the device has run a full shortest path first (SPF) calculation.
Step 8	**show isis** [*process-tag*] **topology** **Example:** `Device# show isis financetag topology`	Displays a list of all connected devices in all areas. • If a process tag is specified, output is limited to the specified routing process. When "null" is specified for the process tag, the output is displayed only for the device process that has no tag specified. If a process tag is not specified, the output is displayed for all processes.

Examples

The following sample output from the **show isis spf-log** command displays this information:

- When the SPFs were executed
- Total elapsed time for the SPF computation
- Number of nodes that make up the topology in the SPF calculation
- Number of triggers that caused the SPF calculation
- Information regarding what triggered the SPF calculation

```
Device# show isis spf-log

   Level 1 SPF log
When      Duration  Nodes  Count     Last trigger LSP Triggers
00:15:46    3124     40      1          milles.00-00  TLVCODE
00:15:24    3216     41      5          milles.00-00  TLVCODE NEWLSP
00:15:19    3096     41      1          deurze.00-00  TLVCODE
00:14:54    3004     41      2          milles.00-00  ATTACHFLAG LSPHEADER
00:14:49    3384     41      1          milles.00-01  TLVCODE
00:14:23    2932     41      3          milles.00-00  TLVCODE
00:05:18    3140     41      1                        PERIODIC
00:03:54    3144     41      1          milles.01-00  TLVCODE
00:03:49    2908     41      1          milles.01-00  TLVCODE
00:03:28    3148     41      3           bakel.00-00  TLVCODE TLVCONTENT
00:03:15    3054     41      1          milles.00-00  TLVCODE
00:02:53    2958     41      1          mortel.00-00  TLVCODE
```

Configuration Examples for IS-IS Support for Route Tags

Example Assigning a High Priority Tag Value to an IS-IS IP Prefix

The following example uses the **ip route priority high** command to assign a tag value of 200 to the IS-IS IP prefix:

```
interface Ethernet 0
 ip router isis
 isis tag 200
!
router isis
 ip route priority high tag 200
```

Example Tagging Routes for Networks Directly Connected to an Interface and Redistributing Them

In this example, two interfaces are tagged with different tag values. By default, these two IP addresses would have been put into the IS-IS Level 1 and Level 2 database. However, by using the **redistribute** command with a route map to match tag 110, only IP address 172.16.10.5 255.255.255.0 is put into the Level 2 database.

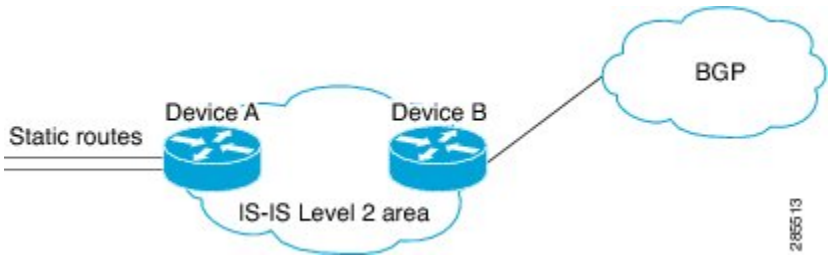
```
interface ethernet 1/0
 ip address 192.168.129.1 255.255.255.0
 ip router isis
 isis tag 120
interface ethernet 1/1
 ip address 172.16.10.5 255.255.255.0
 ip router isis
 isis tag 110
router isis
 net 49.0001.0001.0001.0001.00
 redistribute isis ip level-1 into level-2 route-map match-tag
route-map match-tag permit 10
 match tag 110
```

Example: Redistributing IS-IS Routes Using a Route Map

In a scenario using route tags, you might configure some commands on one device and other commands on another device. For example, you might have a route map that matches on a tag and sets a different tag on a device at the edge of a network, and on different devices you might configure the redistribution of routes based on a tag in a different route map.

The figure below illustrates a flat Level 2 Intermediate System-to-Intermediate System (IS-IS) area. On the left edge are static routes from Device A to reach some IP prefixes. Device A redistributes the static routes into IS-IS. Device B runs the Border Gateway Protocol (BGP) and redistributes IS-IS routes into BGP and then uses the tag to apply different administrative policy based on different tag values.

Figure 146: Example of Redistributing IS-IS Routes Using a Route Map



Device A

```
router isis
 net 49.0000.0000.0001.00
 metric-style wide
 redistribute static ip route-map set-tag
!
route-map set-tag permit 5
 set tag 10
```

Device B

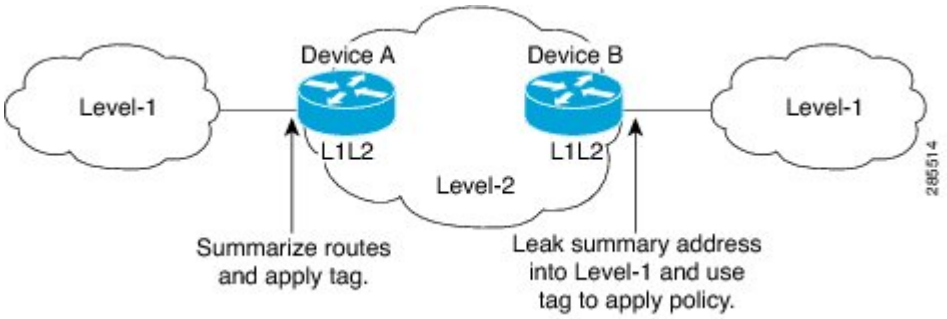
```
router bgp 100
 redistribute isis level-2 route-map tag-policy
route-map tag-policy permit 20
 match tag 10
 set metric 1000
```

Example: Tagging a Summary Address and Applying a Route Map

The figure below illustrates two Level 1 areas and one Level 2 area between them. Device A and Device B are Level 1/Level 2 edge devices in the Level 2 area. On edge Device A, a summary address is configured to reduce the number of IP addresses put into the Level 2 Intermediate System-to-Intermediate System (IS-IS) database. Also, a tag value of 100 is set to the summary address.

On Device B, the summary address is leaked into the Level 1 area, and administrative policy is applied based on the tag value.

Figure 147: Tag on a Summary Address



Device A

```
router isis
 net 49.0001.0001.0001.00
 metric-style wide
 summary-address 10.0.0.0 255.0.0.0 tag 100
```

Device B

```
router isis
 net 49.0002.0002.0002.0002.0
 metric-style wide
 redistribute isis ip level-2 into level-1 route-map match-tag
route-map match-tag permit 10
 match tag 100
```

Example Filtering and Redistributing IS-IS Routes Using an Access List and a Route Map

In this example, the first **redistribute isis ip** command controls the redistribution of Level 1 routes into Level 2. Only the routes with the tag of 90 and whose IP prefix is not 192.168.130.5/24 will be redistributed from Level 1 into Level 2.

The second **redistribute isis ip** command controls the route leaking from Level 2 into the Level 1 domain. Only the routes tagged with 60 or 50 will be redistributed from Level 2 into Level 1.

```
interface ethernet 1
 ip address 192.168.130.5 255.255.255.0
 ip router isis
 isis tag 60
```

```
!
interface ethernet 2
 ip address 192.168.130.15 255.255.255.0
 ip router isis
 isis tag 90
!
interface ethernet 3
 ip address 192.168.130.25 5 255.255.255.0
 ip router isis
 isis tag 50
!
router isis
 net 49.0001.0001.0001.0001.00
 metric-style wide
 redistribute isis ip level-1 into level-2 route-map redist1-2
 redistribute isis ip level-2 into level-1 route-map leak2-1
!
access-list 102 deny ip host 192.168.130.5 host 255.255.255.255
access-list 102 permit ip any any
!
route-map leak2-1 permit 10
 match tag 60
!
route-map leak2-1 permit 20
 match tag 50
!
route-map redist1-2 permit 10
 match ip address 102
 match tag 90
```

Example: IS-IS Limit on the Number of Redistributed Routes

This example shows how to set a maximum of 1200 prefixes that can be redistributed into an Intermediate System-to-Intermediate System (IS-IS). When the number of prefixes redistributed reaches 80 percent of 1200 (960 prefixes), a warning message is logged. When 1200 prefixes are redistributed, IS-IS rebuilds the link-state packet (LSP) fragments without external prefixes and no redistribution occurs.

```
router isis 1
 redistribute maximum-prefix 1200 80 withdraw
```

Example: Requesting a Warning About the Number of Redistributed Routes

This example shows how to allow two warning messages to be logged. The first message is generated if the number of prefixes redistributed reaches 85 percent of 600 (510 prefixes), and the second message is generated if the number of redistributed prefixes reaches 600. However, the number of redistributed prefixes is not limited. If the LSPFULL state occurs, external prefixes are suppressed.

```
router isis 1
 redistribute maximum-prefix 600 85 warning-only
 lsp-full suppress external
```

Example Excluding Connected IP Prefixes on a Small Scale

The following example uses the **no isis advertise-prefix** command on Ethernet interface 0. Only the IP address of loopback interface 0 is advertised.

```
!
interface loopback 0
 ip address 192.168.10.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet 0
 ip address 192.168.20.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
 no isis advertise-prefix
.
.
.
router isis
 passive-interface loopback 0
 net 47.0004.004d.0001.0001.0c11.1111.00
 log-adjacency-changes
!
```

Example Excluding Connected IP Prefixes on a Large Scale

The following example uses the **advertise-passive-only** command, which applies to the entire IS-IS instance, thereby preventing IS-IS from advertising the IP network of Ethernet interface 0. Only the IP address of loopback interface 0 is advertised.

```
!
interface loopback 0
 ip address 192.168.10.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet0
 ip address 192.168.20.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
.
.
.
router isis
 passive-interface Loopback0
 net 47.0004.004d.0001.0001.0c11.1111.00
 advertise-passive-only
 log-adjacency-changes
!
```

Where to Go Next

To configure features to improve Intermediate System-to-Intermediate System (IS-IS) network convergence times, complete the optional tasks in one or more of the following modules in the *IP Routing: IS-IS Configuration Guide*:

- “Overview of IS-IS Fast Convergence”
- “Reducing Failure Detection Times in IS-IS Networks”
- “Reducing Link Failure and Topology Change Notification Times in IS-IS Networks”

Additional References

Related Documents

Related Topic	Document Title
Description of IS-IS type length value (TLV) and its use.	Intermediate System-to-Intermediate Systems (IS-IS) TLVs
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
IS-IS route leaking	IS-IS Route Leaking
Overview of Cisco IS-IS conceptual information with links to all the individual IS-IS modules	"Integrated IS-IS Routing Protocol Overview" module

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Reducing Alternate-Path Calculation Times in IS-IS Networks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 171

Enhancing Security in an IS-IS Network

This module describes processes that you can follow to enhance network security when you use Intermediate System-to-Intermediate System (IS-IS) in your network. You can set passwords, prevent unauthorized routers from forming adjacencies with routers in your IS-IS network, and use the IS-IS HMAC-MD5 Authentication and Enhanced Clear Text Authentication feature.

Prerequisites for Enhancing Security in an IS-IS Network

- Before performing the tasks in this module, you should be familiar with the concepts described in the "Integrated IS-IS Routing Protocol Overview" and "Configuring a Basic IS-IS Network" modules.
- It is assumed you already have IS-IS running on your network.

Information About Enhancing Security in an IS-IS Network

Importance of Preventing Unauthorized Information from Entering an IS-IS Network

It is recommended that you configure the security features described in this module in order to prevent unauthorized routing messages from being placed into the network routing domain. You can set an authentication password for each interface, as well as set an area password for each IS-IS area to prevent unauthorized devices from injecting false routing information into the link-state database, or you can configure a type of IS-IS authentication--either IS-IS HMAC-MD5 or enhanced clear text authentication.

The following sections describe configuration tasks for IS-IS authentication. Two types of authentication are supported: IS-IS HMAC-MD5 and clear text. The task you perform depends on whether you are introducing authentication or migrating from an existing authentication scheme.

Before you can configure authentication, you must make the following decisions:

- Whether to configure authentication for the IS-IS instance and/or for individual IS-IS interfaces (both tasks are included in this section).
- At what level(s) authentication is to be used.
- What type of authentication (IS-IS HMAC-MD5 or clear text) is to be used.

IS-IS Authentication Functionality

New style IS-IS authentication (IS-IS HMAC-MD5 and clear text) provides a number of advantages over the old style password configuration commands that were described in the previous sections, "Setting an Authentication Password for each Interface" and "Setting a Password at Level 1".

- Passwords are encrypted when the software configuration is displayed.
- Passwords are easier to manage and change.
- Passwords can be rolled over to new passwords without disrupting network operations.
- Non-disruptive authentication transitions are supported by allowing configuration which allowed the router to accept PDUs without authentication or with stale authentication information, yet send PDUs with current authentication. Such transitions are useful when you are migrating from no authentication to some type of authentication, when you are changing authentication type, and when you are changing keys.

IS-IS has five PDU types: link state PDU (LSP), LAN Hello, Point-to-Point Hello, complete sequence number PDU (CSNP), and partial sequence number PDU (PSNP). IS-IS HMAC-MD5 authentication or clear text password authentication can be applied to all five PDU types. The authentication can be enabled on different IS-IS levels independently. The interface-related PDUs (LAN Hello, Point-to-Point Hello, CSNP, and PSNP) can be enabled with authentication on different interfaces, with different levels and different passwords.

Benefits of IS-IS Clear Text Authentication

IS-IS clear text (plain text) authentication provides the same functionality as is provided by using the **area-password** or **domain-password** command. However, use of clear text authentication takes advantage of the more flexible key management capabilities described above.

Benefits of IS-IS HMAC-MD5 Authentication

- IS-IS now supports MD5 authentication, which is more secure than clear text authentication. IS-IS HMAC-MD5 authentication adds an HMAC-MD5 digest to each IS-IS protocol data unit (PDU). HMAC is a mechanism for message authentication codes (MACs) using cryptographic hash functions. The digest allows authentication at the IS-IS routing protocol level, which prevents unauthorized routing messages from being injected into the network routing domain.
- MD5 authentication or clear text authentication can be enabled on Level 1 or Level 2 independently.
- Passwords can be rolled over to new passwords without disrupting routing messages.
- For the purpose of network transition, you can configure the networking device to accept PDUs without authentication or with wrong authentication information, yet send PDUs with authentication. Such transition might be because you are migrating from no authentication to some type of authentication, you are changing authentication type, or you are changing keys.

Before you migrate from using one type of security authentication to another, all routers must be loaded with the new image that supports the new authentication type. The routers will continue to use the original authentication method until all routers have been loaded with the new image that supports the new authentication method, and all routers have been configured to use the new authentication method. Once all routers are loaded with the required image, you must follow the configuration steps for the desired new authentication method as described in the previous Configuring HMAC-MD5 or Clear Text Authentication for an IS-IS Interface, on page 2201 . You also must decide whether to configure authentication for the IS-IS area or for individual IS-IS interfaces. Both tasks are included in the referenced section.

Note To achieve a smooth transition from one authentication method to another, allowing for continuous authentication of IS-IS PDUs, perform the task steps in the order shown, which requires moving from router to router doing certain steps before all the steps are performed on any one router.

Migration from Old Clear Text Authentication to HMAC-MD5 Authentication

When you configure MD5 authentication, the **area-password** and **domain-password** command settings will be overridden automatically with the new authentication commands. When you configure MD5 authentication, the **isis password** command setting will be overridden automatically with the new authentication commands.

Migration from Old Clear Text Authentication to the New Clear Text Authentication

The benefits of migrating from the old method of clear text authentication to the new method of clear text authentication are as follows:

- Passwords are easier to change and maintain.
- Passwords can be encrypted when the system configuration is being displayed (if you use key management).

ISIS Authentication Changes

ISIS supports both plain text and cryptographic authentication. However, only one authentication scheme can be configured at a time:

- Configure plain text authentication using the **area-password** command.
- Configure cryptographic authentication using the **authentication key-chain** command for MD5, SHA, or other authentication schemes.

The following behavioral change was introduced that impacts the ISIS authentication configuration method.

Starting with Release 16.10.1, the **authentication key-chain** command can be used to enable cryptographic authentication. Therefore, plain text authentication cannot be configured using the **area-password** command if the **authentication key-chain** command is already configured.

After Release 16.10.1, you are no longer required to issue the **authentication mode** command. Enter the **authentication key-chain** command to configure cryptography. This command cannot co-exist with the plain-text **area-password** command. As a result of the new behavior, you will see the following error message when you attempt to configure authentication in combination with the **authentication key-chain** command:

```
Device#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)#router isis abc
Device(config-router)#authentication key-chain isis-key
```

```
Device(config-router)#area-password text-pw
%Please configure password using authentication command
Device(config-router)
```

Since the new software does not allow configuration of the **authentication key-chain** command to coexist with the **area-password** command, the behavior change will cause a service interruption when a device is upgraded. This command will be automatically deleted from the new configuration.

How to Enhance Security in an IS-IS Network

Setting an Authentication Password for each Interface

Note The password is exchanged as plain text and thus provides only limited security.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **isis password** *password* [**level-1**| **level-2**]
5. Repeat Step 4 for each interface password that you want to set.
6. **end**
7. **show ip interface** [*type number*] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Enters interface configuration mode.
Step 4	**isis password** *password* [**level-1**| **level-2**] **Example:**	Configures the authentication password for an interface.

	Command or Action	Purpose
	`Device(config-if)# isis password sjpass level-1`	• Different passwords can be assigned for different routing levels using the **level-1** and **level-2** keywords. • Specifying the **level-1** or **level-2** keyword disables the password only for Level 1 or Level 2 routing, respectively.
Step 5	Repeat Step 4 for each interface password that you want to set.	--
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show ip interface** [*type number*] [**brief**] **Example:** `Device# show ip interface gigabitethernet 0/0/0`	Displays the usability status of interfaces configured for IP.

Setting a Password at Level 1

Note This password is exchanged as plain text, and, thus, this feature provides only limited security.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [**area-** *tag*]
4. **area-password** *password*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router isis** [**area-** *tag*] **Example:** `Device(config)# router isis salesarea`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**area-password** *password* **Example:** `Device(config-router)# area-password companyz`	Configures the IS-IS area authentication password. • Using the **area-password** command on all devices in an area will prevent unauthorized devices from injecting false routing information into the link-state database. • This password is inserted in Level 1 protocol data unit (PDU) link-state PDUs (LSPs), complete sequence number PDUs (CSNPs), and partial sequence number PDUs (PSNPs).
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Setting a Password at Level 2

Note This password is exchanged as plain text, and, thus, this feature provides only limited security.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **domain-password** *password* [**authenticate snp** {**validate** | **send-only**}]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis salesarea`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 4	**domain-password** *password* [**authenticate snp** {**validate** | **send-only**}] **Example:** `Device(config-router)# domain-password company2`	Configures the IS-IS routing domain authentication password. **Note** If you do not specify the **authenticate snp** keyword along with either the **validate** or **send-only** keyword, the IS-IS routing protocol does not insert the password into SNPs. **Note** Using the **domain-password** command on all devices in an area will prevent unauthorized devices from injecting false routing information into the link-state database. **Note** This password is inserted in Level 2 PDU link-state PDUs (LSPs), complete sequence number PDUs (CSNPs), and partial sequence number PDUs (PSNPs). If you specify the **authenticate snp** keyword along with either the **validate** or **send-only** keyword, the IS-IS routing protocol will insert the password into sequence number PDUs (SNPs).
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Configuring IS-IS Authentication

Configuring HMAC-MD5 Authentication or Clear Text Authentication for the First Time

Configuring HMAC-MD5 or Clear Text Authentication for the IS-IS Instance

Before you begin

In order to use HMAC-MD5 or clear text authentication with encrypted keys, the Integrated IS-IS routing protocol must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *text*
6. **exit**
7. **exit**
8. **router isis** [**area-** *tag*]
9. **authentication send-only** [**level-1** | **level-2**]
10. Repeat Steps 1 through 9 on each device that will communicate.
11. **authentication key-chain** *name-of-chain* [**level-1** | **level-2**]
12. Repeat Steps 11 and 12 on each router that will communicate.
13. **no authentication send-only**
14. Repeat Step 14 on each device that will communicate.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain remote3754`	Enables authentication for routing protocols and identifies a group of authentication keys.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 100`	Identifies an authentication key on a key chain. • The *key-id* argument must be a number.
Step 5	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string mno172`	Specifies the authentication string for a key. • The *text* argument can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a number.
Step 6	**exit** **Example:**	Returns to keychain configuration mode.

	Command or Action	Purpose
	`Device(config-keychain-key)# exit`	
Step 7	**exit** **Example:** `Device(config-keychain)# exit`	Returns to global configuration mode.
Step 8	**router isis** [**area-** *tag*] **Example:** `Device(config)# router isis 1`	Enables IS-IS as an IP routing protocol and assigns a tag to a process, if required. • Enters router configuration mode.
Step 9	**authentication send-only** [**level-1** | **level-2**] **Example:** `Device(config-router)# authentication send-only`	Specifies for the IS-IS instance that MD5 authentication is performed only on IS-IS PDUs being sent (not received).
Step 10	Repeat Steps 1 through 9 on each device that will communicate.	Use the same key string on each device.
Step 11	**authentication key-chain** *name-of-chain* [**level-1** | **level-2**] **Example:** `Device(config-router)# authentication key-chain remote3754`	Enables MD5 authentication for the IS-IS instance.
Step 12	Repeat Steps 11 and 12 on each router that will communicate.	--
Step 13	**no authentication send-only** **Example:** `Device(config-router)# no authentication send-only`	Specifies for the IS-IS instance that MD5 authentication is performed on IS-IS PDUs being sent and received. • In Step 9 you enable authentication to be performed only for IS-IS PDUs that are being sent. In Step 14 you enter the **no authentication send-only** command so that the authentication is now performed on PDUs sent and received.
Step 14	Repeat Step 14 on each device that will communicate.	--

Configuring HMAC-MD5 or Clear Text Authentication for an IS-IS Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*

5. **key-string** *text*
6. **exit**
7. **exit**
8. **interface** *type number*
9. **isis authentication send-only** [**level-1** | **level-2**]
10. Repeat Steps 1 through 9 on each device that will communicate.
11. **isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**]
12. Repeat Steps 11 and 12 on each router that will communicate.
13. **no isis authentication send-only**
14. Repeat Step 14 on each device that will communicate.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain multistate87723`	Enables authentication for routing protocols and identifies a group of authentication keys.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 201`	Identifies an authentication key on a key chain. • The *key-id* argument must be a number.
Step 5	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string idaho`	Specifies the authentication string for a key. • The *text* argument can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a number.
Step 6	**exit** **Example:** `Device(config-keychain-key)# exit`	Returns to keychain configuration mode.
Step 7	**exit** **Example:**	Returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-keychain)# exit`	
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface.
Step 9	**isis authentication send-only** [**level-1** | **level-2**] **Example:** `Device(config-if)# isis authentication send-only`	Specifies that authentication is performed only on PDUs being sent (not received) on a specified IS-IS interface.
Step 10	Repeat Steps 1 through 9 on each device that will communicate.	Use the same key string on each device.
Step 11	**isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**] **Example:** `Device(config-if)# isis authentication key-chain multistate87723`	Enables MD5 authentication for an IS-IS interface.
Step 12	Repeat Steps 11 and 12 on each router that will communicate.	--
Step 13	**no isis authentication send-only** **Example:** `Device(config-if)# no isis authentication send-only`	Specifies that authentication is performed on PDUs being sent and received on a specified IS-IS interface.
Step 14	Repeat Step 14 on each device that will communicate.	--

Migrating to a New Authentication Type

SUMMARY STEPS

1. Load all devices with the image required to support the new, desired authentication method.
2. Configure the new authentication mode on both the interface and the IS-IS area by following the appropriate tasks in the Configuring HMAC-MD5 Authentication or Clear Text Authentication for the First Time, on page 2199.

DETAILED STEPS

Step 1 Load all devices with the image required to support the new, desired authentication method.

Step 2 Configure the new authentication mode on both the interface and the IS-IS area by following the appropriate tasks in the Configuring HMAC-MD5 Authentication or Clear Text Authentication for the First Time, on page 2199.

Configuring Authentication on a New Router Being Added to a Network That Already Has Authentication Configured

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *text*
6. **exit**
7. **exit**
8. **interface** *type number*
9. **isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain multistate87723`	Enables authentication for routing protocols and identifies a group of authentication keys.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 201`	Identifies an authentication key on a key chain. • The *key-id* argument must be a number.
Step 5	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string idaho`	Specifies the authentication string for a key. • The *text* argument can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a number.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Device(config-keychain-key)# exit`	Returns to keychain configuration mode.
Step 7	**exit** **Example:** `Device(config-keychain)# exit`	Returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface.
Step 9	**isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**] **Example:** `Device(config-if)# isis authentication key-chain multistate87723`	Enables MD5 authentication for an IS-IS interface.

Configuration Examples for Enhancing Security in an IS-IS Network

Example Configuring IS-IS HMAC-MD5 Authentication

The following example configures a key chain and key for IS-IS HMAC-MD5 authentication for GigabitEthernet interface 3/0/0 (on Hello PDUs) and for the IS-IS instance (on LSP, CSNP, and PSNP PDUs).

```
!
key chain cisco
 key 100
 key-string tasman-drive
!
interface GigabitEthernet3/0/0
 ip address 10.1.1.1 255.255.255.252
 ip router isis real_secure_network
 isis authentication key-chain cisco level-1
!
router isis real_secure_network
 net 49.0000.0101.0101.0101.00
 is-type level-1
 authentication key-chain cisco level-1
!
```

Example Configuring IS-IS Clear Text Authentication

The following example configures a key chain and key for IS-IS clear text authentication for GigabitEthernet interface 3/0/0 (on Hello PDUs) and for the IS-IS instance (on LSP, CSNP, and PSNP PDUs).

```
!
key chain cisco
 key 100
 key-string tasman-drive
!
interface GigabitEthernet3/0/0
 ip address 10.1.1.1 255.255.255.252
 ip router isis real_secure_network
 isis authentication key-chain cisco level-1
!
router isis real_secure_network
 net 49.0000.0101.0101.0101.00
 is-type level-1
 authentication key-chain cisco level-1
!
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
IPv6 Routing: IS-IS Multitopology Support for IPv6	" *Reducing Link Failure and Topology Change Notification Times in IS-IS Networks* " module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Enhancing Security in an IS-IS Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 172

IS-IS IPv6 Administrative Tag

The IS-IS IPv6 Administrative Tag feature allows you to assign a tag to IPv6 prefixes that you can use to apply administrative policies with a route map. For example, you can control routes redistributed between area and domain boundaries and between different routing protocols, or apply policies on Intermediate System-to-Intermediate System (IS-IS) routes.

Information About IS-IS IPv6 Administrative Tag

IS-IS Administrative Tags in IPv6 Prefixes

You can configure an IS-IS administrative tag value for IPv6 prefixes. You can then specify the tag value of IPv6 prefixes that IS-IS inserts into the link-state protocol data units (PDUs) it generates and those that it retrieves from LSPs.

How to Configure an IS-IS IPv6 Administrative Tag

Assigning a Tag to an IS-IS IPv6 Prefix

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **net** *net1*
5. **metric-style wide**
6. **interface** [*type number*]

7. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **ipv6 router isis** [*area-tag*]
9. **isis ipv6 tag** *tag-value*
10. **end**
11. **show isis database verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis tag13`	Enables the IS-IS routing protocol, specifies an IS-IS process, and enters router configuration mode.
Step 4	**net** *net1* **Example:** `Device(config-router)# net 49.0000.0000.0100.00`	Configures an IS-IS network entity table (NET) for the routing process.
Step 5	**metric-style wide** **Example:** `Device(config-router)# metric-style wide`	Configures a router running IS-IS so that it generates and accepts only new-style type, length, value objects (TLVs).
Step 6	**interface** [*type number*] **Example:** `Device(config-router)# interface GigabitEthernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 7	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2005::1/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

	Command or Action	Purpose
Step 8	**ipv6 router isis** [*area-tag*] **Example:** `Device(config-if)# ipv6 router isis area1`	Configures an IS-IS routing process for IPv6 on an interface and attaches an area designator to the routing process.
Step 9	**isis ipv6 tag** *tag-value* **Example:** `Device(config-if)# isis ipv6 tag 200`	Configures an administrative tag value that will be associated with an IPv6 address prefix and applied to an IS-IS LSP.
Step 10	**end** **Example:** `Device(config-if)# end`	(Optional) Saves configuration commands to the running configuration file and returns to privileged EXEC mode.
Step 11	**show isis database verbose** **Example:** `Device# show isis database verbose`	(Optional) Displays details about the IS-IS link-state database, including the route tag. • Enter this command if you want to verify the tag.

Assigning a High Priority Administrative Tag to an IS-IS IPv6 Prefix

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **address-family ipv6**
5. **ipv6 route priority high tag** *tag-value*
6. **exit**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Configures an IS-IS routing process for IP on an interface, attaches an area designator to the routing process, and enters router configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode.
Step 5	**ipv6 route priority high tag** *tag-value* **Example:** `Device(config-router-af)# ipv6 route priority high tag 200`	Assigns a high priority tag to an IS-IS IPv6 prefix.
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	(Optional) Exits address family configuration mode, and returns to router configuration mode.
Step 7	**exit** **Example:** `Device(config-router)# exit`	(Optional) Exits router configuration mode, and returns to global configuration mode.

Using an IS-IS IPv6 Administrative Tag to Redistribute Routes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **address-family ipv6**
5. **redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} [**distribute-list** *list-name*] [**route-map** *map-tag*]
6. **exit**
7. **exit**
8. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
9. **match tag** *tag-value* [*...tag-value*]
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Configures an IS-IS routing process for IP on an interface, attaches an area designator to the routing process, and enters router configuration mode.
Step 4	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode.
Step 5	**redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} [**distribute-list** *list-name*] [**route-map** *map-tag*] **Example:** `Device(config-router-af)# redistribute isis level-1 into level-2 route-map IPV6-PERMIT-TAG`	Redistributes IPv6 routes from one routing domain into another routing domain using IS-IS as both the target and source protocol.
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	(Optional) Exits address family configuration mode, and returns to router configuration mode.
Step 7	**exit** **Example:** `Device(config-router)# exit`	(Optional) Exits router configuration mode, and returns to global configuration mode.
Step 8	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map match-tag`	Defines the conditions for redistributing routes from one routing protocol into another or from one IS-IS level to another. • This command causes the router to enter route-map configuration mode.
Step 9	**match tag** *tag-value* [*...tag-value*]	Matches routes tagged with the specified tag numbers.

	Command or Action	Purpose
	Example: `Device(config-route-map)# match tag 100`	• If you are setting a tag for the first time, you cannot match on tag; this step is an option if you are changing tags.
Step 10	**exit** **Example:** `Device(config-route-map)# exit`	(Optional) Exits route-map configuration mode, and returns to global configuration mode.

Using an IS-IS IPv6 Administrative Tag to Configure Routes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]
4. **router isis** [*area-tag*]
5. **address-family ipv6**
6. **redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} [**distribute-list** *list-name*] [**route-map** *map-tag*
7. **exit**
8. **exit**
9. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
10. **set tag** *tag-value*
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address*	*interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name*	**default**]] [*administrative-distance*]	Establishes a static IPv6 routes.

	Command or Action	Purpose
	[*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **Example:** `Router(config)# ipv6 route  2033::1/64 GigabitEthernet 0/0/0`	
Step 4	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Configures an IS-IS routing process for IP on an interface, attaches an area designator to the routing process, and enters router configuration mode.
Step 5	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode.
Step 6	**redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} [**distribute-list** *list-name*] [**route-map** *map-tag* **Example:** `Device(config-router-af)# redistribute isis level-1 into level-2 route-map IPV6-PERMIT-TAG`	Redistributes IPv6 routes from one routing domain into another routing domain using IS-IS as both the target and source protocol.
Step 7	**exit** **Example:** `Device(config-router-af)# exit`	(Optional) Exits address family configuration mode, and returns to router configuration mode.
Step 8	**exit** **Example:** `Device(config-router)# exit`	(Optional) Exits router configuration mode, and returns to global configuration mode.
Step 9	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map set-tag`	Defines the conditions for redistributing routes from one routing protocol into another or from one IS-IS level to another. • This command causes the router to enter route-map configuration mode.
Step 10	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 300`	Sets a tag value of the destination routing protocol.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Device(config-route-map)# exit`	(Optional) Exits route-map configuration mode, and returns to global configuration mode.

Applying an IS-IS IPv6 Tag to a Summary Prefix

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **address-family ipv6**
5. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]
6. **exit**
7. **exit**
8. **router isis** [*area-tag*]
9. **address-family ipv6**
10. **redistribute isis** [*process-id*] {**level-1**| **level-2**} **into** {**level-1** |**level-2**} **distribute-list** *list-name*
11. **summary-prefix** *ipv6-prefix/ prefix-length* {**level-1** | **level-1-2**| **level-2**} **tag** *tag-value*
12. **end**
13. **show isis database verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Configures an IS-IS routing process for IP on an interface, attaches an area designator to the routing process, and enters router configuration mode.

	Command or Action	Purpose
Step 4	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode.
Step 5	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **Example:** `Device(config-router-af)# ipv6 route 11:1:1:1:1:1::/96 GigabitEthernet 0/0/0`	Establishes a static IPv6 routes.
Step 6	**exit** **Example:** `Device(config-router-af)# exit`	(Optional) Exits address family configuration mode, and returns to router configuration mode.
Step 7	**exit** **Example:** `Device(config-router)# exit`	(Optional) Exits router configuration mode, and returns to global configuration mode.
Step 8	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Configures an IS-IS routing process for IP on an interface, attaches an area designator to the routing process, and enters router configuration mode.
Step 9	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode.
Step 10	**redistribute isis** [*process-id*] {**level-1**| **level-2**} **into** {**level-1** |**level-2**} **distribute-list** *list-name* **Example:** `Device(config-router-af)# redistribute static level-2 metric 50`	Redistributes IPv6 routes from one routing domain into another routing domain using IS-IS as both the target and source protocol.
Step 11	**summary-prefix** *ipv6-prefix/ prefix-length* {**level-1** | **level-1-2**| **level-2**} **tag** *tag-value* **Example:**	Allows a Level 1-2 router to summarize Level 1 prefixes at Level 2, instead of advertising the Level 1 prefixes directly when the router advertises the summary. • The *ipv6-prefix* argument in the **summary-prefix** command must be in the form documented in RFC

	Command or Action	Purpose
	`Device(config-router-af)# summary-prefix 11:1:1:1::/64 tag 600`	2373 where the address is specified in hexadecimal using 16-bit values between colons. • The *prefix-length* argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 12	**end** **Example:** `Device(config-router-af)# end`	(Optional) Saves configuration commands to the running configuration file and returns to privileged EXEC mode.
Step 13	**show isis database verbose** **Example:** `Device# show isis database verbose`	(Optional) Displays details about the IS-IS link-state database, including the route tag. • Perform this step if you want to verify the tag.

Configuration Examples for IS-IS IPv6 Administrative Tag

Example: Assigning a Tag to an IS-IS IPv6 Prefix

```
Device(config)# router isis
Device(config-router)# net 49.0000.0000.0100.00
Device(config-router)# metric-style wide
Device(config-router)# interface GigabitEthernet 0/0/0
Device(config-if)# ipv6 address 2005::1/64
Device(config-if)# ipv6 router isis
Device(config-if)# isis ipv6 tag 200
Device(config-if)# end
Device# show isis database verbose

IS-IS Level-1 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Device.00-00     * 0x00000001   0xD27D        1189              0/0/0
  Area Address: 49
  NLPID:        0x8E
  Hostname: Device
  IPv6 Address: 2005::1
  Metric: 10         IPv6 2005::/64
    Route Admin Tag: 200
IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Device.00-00     * 0x00000001   0xD27D        1189              0/0/0
  Area Address: 49
  NLPID:        0x8E
  Hostname: Device
  IPv6 Address: 2005::1
  Metric: 10         IPv6 2005::/64
```

```
    Route Admin Tag: 200
```

Example: Assigning a High Priority Administrative Tag to an IS-IS IPv6 Prefix

```
Device(config)# router isis
Device(config-router)# address-family ipv6
Device(config-router-af)# ipv6 route priority high tag 200
```

Example: Using an IS-IS IPv6 Administrative Tag to Redistribute Routes

```
Device(config)# router isis
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute isis level-1 into level-2 route-map match-tag
Device(config-router-af)# route-map match-tag
Device(config-route-map)# match tag 100
```

Example: Using an IS-IS IPv6 Administrative Tag to Configure Routes

```
Device(config)# ipv6 route 2033::1/64 GigabitEthernet 0/0/0
Device(config)# router isis
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute static level-2 route-map set-tag
Device(config-router-af)# route-map set-tag
Device(config-route-map)# set tag 300
Device(config-route-map)# end
Device# show isis database verbose level-2

Device.00-00     * 0x0000004E   0x9805        1197              0/0/0
  Area Address: 33
  NLPID:        0xCC 0x8E
  Hostname: Device
  IP Address:   10.100.100.20
  IPv6 Address: 2001:DB8::100
  IPv6 Address: 2001:DB8::200
  Metric: 10         IS-Extended route500.01
  Metric: 10         IP 10.100.100.0/24
  Metric: 10         IPv6 2001:DB8::/64
  Metric: 10         IPv6 2001:DB8::/64
  Metric: 10         IPv6-Interarea 11:1:1:1:1:1:1:1/128
  Metric: 20         IPv6-Interarea 2003:DB8::/64
  Metric: 0          IPv6 2033::/64
    Route Admin Tag: 300
```

Example: Applying an IS-IS IPv6 Administrative Tag to a Summary Prefix

```
Device(config)# router isis
Device(config)# ipv6 route 11:1:1:1:1:1::/96 GigabitEthernet 0/0/0
Device(config)# router isis
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute static level-2 metric 50
Device(config-router-af)# summary-prefix 11:1:1:1::/64 tag 600
Device(config-route-map)# end
Device# show isis database verbose level-2

IS-IS Level-2 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
Device.00-00     * 0x00000007   0x4AA7        1174              0/0/0
  Area Address: 33
  NLPID:        0xCC 0x8E
  Hostname: Device
  IP Address:   10.100.100.20
  IPv6 Address: 2001:DB8::100
  IPv6 Address: 2001:DB8::200
  Metric: 10         IS-Extended route500.01
  Metric: 10         IP 10.100.100.0/24
  Metric: 10         IPv6 2001:DB8::/64
  Metric: 10         IPv6 2001:DB8::/64
  Metric: 10         IPv6 11:1:1:1::/64
    Route Admin Tag: 600
(Summary route 11:1:1:1::/64 is advertised with tag 600)
Device(config-router-af)#
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IS-IS IPv6 Administrative Tag

CHAPTER 173

IS-IS IPv6 Advertise Passive Only

The IS-IS IPv6 Advertise Passive Only feature allows you to configure the Intermediate System-to-Intermediate System (IS-IS) instance on a device to advertise only IPv6 prefixes that belong to passive interfaces and exclude other connected IPv6 prefixes.

Prerequisites for IS-IS IPv6 Advertise Passive Only

Before you can use the IS-IS IPv6 Advertise Passive Only feature to exclude IPv6 prefixes of connected networks from IS-IS link-state protocol (LSP) data unit advertisements, the integrated IS-IS routing protocol must be configured. See the "Configuring a Basic IS-IS Network" section of the *IP Routing: ISIS Configuration Guide*.

Information About IS-IS IPv6 Advertise Passive Only

IPv6 Prefixes Only Allowed on Passive Interfaces

You can configure the IS-IS instance on a device to allow only IPv6 prefixes that belong to passive interfaces in its LSP advertisements. This configuration reduces the number of IPv6 prefixes carried in the LSP advertisement.

How to Configure IS-IS IPv6 Advertise Passive Only

Configuring IS-IS Instances on a Device to Advertise Passive Interface IPv6 Prefixes Only

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **net** *net1*
5. **interface loopback** *number*
6. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
7. **exit**
8. **interface** *type number*
9. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
10. **ipv6 router isis** [*area-tag*]
11. **exit**
12. **router isis** [*area-tag*]
13. **passive-interface** [**default**] *type number*
14. **address-family ipv6**
15. **advertise passive-only**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis area1`	Configures an IS-IS routing process for IP on an interface, attaches an area designator to the routing process, and enters router configuration mode.
Step 4	**net** *net1* **Example:**	Configures an IS-IS network entity table (NET) for the routing process.

	Command or Action	Purpose
	`Device(config-router)# net 47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00`	
Step 5	**interface loopback** *number* **Example:** `Device(config-router)# interface loopback 0`	Configures a loopback interface and enters interface configuration mode.
Step 6	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:688:1001:1000::1/128`	Sets a primary IPv6 address for an interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 9	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:688:1001:100A::1/64`	Configures an IPv6 address for the interface.
Step 10	**ipv6 router isis** [*area-tag*] **Example:** `Device(config-if)# ipv6 router isis area1`	Configures an IS-IS routing process for IPv6 on an interface, attaches an area designator to the routing process, and enters router configuration mode.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 12	**router isis** [*area-tag*] **Example:** `Device(config)# router isis area1`	Configures an IS-IS routing process for IP on an interface, attaches an area designator to the routing process, and enters router configuration mode.

	Command or Action	Purpose
Step 13	**passive-interface** [**default**] *type number* **Example:** `Device(config-router)# passive-interface loopback 0`	Disables sending routing updates on an interface.
Step 14	**address-family ipv6** **Example:** `Device(config-router)# address-family ipv6`	Enters address family configuration mode.
Step 15	**advertise passive-only** **Example:** `Device(config-router-af)# advertise passive-only`	Configures IS-IS to advertise only IPv6 prefixes that belong to passive interfaces.
Step 16	**end** **Example:** `Device(config-router-af)# end`	(Optional) Saves the configuration commands to the running configuration file and returns to privileged EXEC mode.

Configuration Examples for IS-IS IPv6 Advertise Passive Only

Example: Configuring IS-IS Instances on a Device to Advertise Only Passive Interfaces

```
Device> enable
Device# configure terminal
Device(config)# router isis area1
Device(config-router)# net 47.0010.0000.0000.0000.0001.0001.1111.1111.1111.00
Device(config-router)# interface loopback 0
Device(config-if)# ipv6 address 2001:688:1001:1000::1/128
Device(config-if)# exit
Device(config)# interface Ethernet 0/0
Device(config-if)# ipv6 address 2001:688:1001:100A::1/64
Device(config-if)# ipv6 router isis area1
Device(config-if)# exit
Device(config)# router isis area1
Device(config-router)# passive-interface loopback 0
Device(config-router)# address-family ipv6
Device(config-router-af)# advertise passive-only
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
IS-IS commands	*Cisco IOS IS-IS Command Reference*
Configuring the integrated IS-IS routing protocol	"Configuring a Basic IS-IS Network" module of the *IP Routing: ISIS Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IS-IS IPv6 Advertise Passive Only

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 174

IS-IS IPv6 Multi-Process Support

The IS-IS IPv6 Multi-Process Support feature enables support for mutual redistribution of IPv6 routes between multiple IS-IS IPv6 instances and allows the IS-IS IPv6 instances to install routes in non-default virtual routing and forwarding (VRF) instances.

Prerequisites for IS-IS IPv6 Multi-Process Support

- You must enable IPv6 unicast routing before ISIS IPv6 configuration.
- You must enable IPv6 on an interface, by assigning an IPv6 address to the interface or by using the **ipv6 enable** command, before associating the interface with an ISIS IPv6 instance.
- You must define a virtual routing and forwarding (VRF) and enable an IPv6 address family in the VRF before associating an ISIS IPv6 instance with that VRF.

Information About IS-IS IPv6 Multi-Process Support

IS-IS IPv6 Multi-Process Support Overview

The IS-IS IPv6 Multi-Process Support feature allows you to create up to 28 IPv6-enabled IS-IS instances and enables these IPv6 instances to be associated with any VRF and not only the default VRF. The device can redistribute IPv6 routes between multiple IPv6 IS-IS instances in the same VRF including the default VRF. The device can also redistribute routes between an IS-IS instance and other routing protocols such as RIP and OSPFv3 operating in the same VRF, including routing protocols in the default VRF.

How to Configure IS-IS IPv6 Multi-Process Support

Configuring IS-IS IPv6 Multi-Process Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv6** [**unicast**]
5. **exit**
6. **exit**
7. **interface** *type number*
8. (Optional) **vrf forwarding** *vrf-name*
9. **ipv6 address** *ipv6-prefix/prefix-length*
10. **ipv6 router isis** *process-tag*
11. **exit**
12. Repeat Step 7 to Step 11 to configure IS-IS routing process and VRFs for IPv6 on different interfaces.
13. **router isis** *process-tag*
14. (Optional) **vrf** *vrf-name*
15. **net** *network-entity-title*
16. **is-type** [**level-1** | **level-1-2** | **level-2-only**]
17. **log-adjacency-changes**
18. **address-family ipv6** [**unicast**]
19. **redistribute source-protocol** [*process-id*] [**route-map** *map-tag*]
20. **redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} **distribute-list** *list-name*
21. **exit**
22. **exit**
23. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
24. **match route-type** {**level-1** | **level-2**}
25. **set metric** *metric-value*
26. **set level** {**level-1** | **level-2** | **level-1-2**}
27. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition v1`	Configure a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode
Step 4	**address-family ipv6** [**unicast**] **Example:** `Device(config-vrf)# address-family ipv6`	Specifies the IPv6 address family, and enters address family configuration mode. **Note** The **unicast** keyword specifies the unicast IPv6 unicast address family. By default, the device is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the **address-family ipv6** command.
Step 5	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits the address family configuration mode and enters VRF configuration mode.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits the VRF configuration mode and enters global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/2`	Configures an interface type and enters interface configuration mode.
Step 8	(Optional) **vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding v1`	Associates a Virtual Routing and Forwarding (VRF) or a virtual network with an interface or subinterface
Step 9	**ipv6 address** *ipv6-prefix/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8::/32`	Sets an IPv6 address for an interface.
Step 10	**ipv6 router isis** *process-tag* **Example:**	Configures an IS-IS routing process for IPv6 on an interface and attaches a tag to the routing process.

	Command or Action	Purpose
	`Device(config-if)# ipv6 router isis v1a`	**Note** The configuration of the interface-mode **ipv6 router isis** command will overwrite the prior configuration on that interface, but only if the new configuration is attempting to change the interface ownership to a different instance that is in the same VRF as the currently configured owner instance. The configuration will be rejected if the attempted change is between two instances that are associated with different VRFs.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits the interface configuration mode and enters global configuration mode.
Step 12	Repeat Step 7 to Step 11 to configure IS-IS routing process and VRFs for IPv6 on different interfaces. **Example:**	--
Step 13	**router isis** *process-tag* **Example:** `Device(config)# router isis v1a`	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 14	(Optional) **vrf** *vrf-name* **Example:** `Device(config-if)# vrf v1`	Associates a Virtual Routing and Forwarding (VRF) instance or a virtual network with an interface or subinterface
Step 15	**net** *network-entity-title* **Example:** `Device(config-router)# net 49.000b.0000.0001.0002.00`	Configures IS-IS network entity title (NET) for a CLNS routing process.
Step 16	**is-type** [**level-1** | **level-1-2** | **level-2-only**] **Example:** `Device(config-router)# is-type level-1`	Configures the routing level for an instance of the IS-IS routing process.
Step 17	**log-adjacency-changes** **Example:** `Device(config-router)# log-adjacency-changes`	Configure the device to send a syslog message when an Open Shortest Path First (OSPF) neighbor goes up or down.
Step 18	**address-family ipv6** [**unicast**] **Example:**	Specifies the IPv6 address family, and enters address family configuration mode.

	Command or Action	Purpose
	`Device(config-router)# address-family ipv6`	**Note** The **unicast** keyword specifies the unicast IPv6 unicast address family. By default, the device is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the **address-family ipv6** command.
Step 19	**redistribute source-protocol** [*process-id*] [**route-map** *map-tag*] **Example:** `Device(config-router-af)# redistribute isis v1a route-map abc`	Specifies the route map that should be checked to filter the importation of routes from this source routing protocol to the current routing protocol.
Step 20	**redistribute isis** [*process-id*] {**level-1** | **level-2**} **into** {**level-1** | **level-2**} **distribute-list** *list-name* **Example:** `Device(config-router-af)# redistribute isis level-1 into level-2 distribute-list xyz`	Redistributes IPv6 routes from one IS-IS level into another IS-IS level. By default, the routes learned by Level 1 instances are redistributed by the Level 2 instance.
Step 21	**exit** **Example:** `Device(config-router-af)# exit`	Exits the address family configuration mode and enters router configuration mode.
Step 22	**exit** **Example:** `Device(config-router)# exit`	Exits the router configuration mode and enters global configuration mode.
Step 23	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map abc permit 10`	Defines the conditions for redistributing routes from one routing protocol into another or from one IS-IS level to another and enters route-map configuration mode.
Step 24	**match route-type** {**level-1** | **level-2**} **Example:** `Device(config-route-map)# match route-type level-1`	Defines the route-type match criterion.
Step 25	**set metric** *metric-value* **Example:** `Device(config-route-map)# set metric 56`	Configures the metric value used to redistribute routes.
Step 26	**set level** {**level-1** | **level-2** | **level-1-2**} **Example:**	Specifies the routing level of routes to be advertised into a specified area of the routing domain.

	Command or Action	Purpose
	`Device(config-route-map)# set level level-2`	
Step 27	**end** **Example:** `Device(config-route-map)# end`	Exits the route-map configuration mode and enters privileged EXEC mode.

Configuration Examples for IS-IS IPv6 Multi-Process Support

Example: IS-IS IPv6 Multi-Process Support Configuration

```
Device> enable
Device# configure terminal
Device(config)# vrf definition v1
Device(config-vrf)# address-family ipv6
Device(config-router-af)# exit
Device(config-router)# exit
Device(config)# interface FastEthernet 0/2
Device(config-if)# ipv6 address 2001:DB8::/32
Device(config-if)# vrf forwarding v1
Device(config-if)# ipv6 router isis v1a
Device(config-if)# exit
Device(config)# interface FastEthernet 0/3
Device(config-if)# ipv6 address 2001:DB8::/48
Device(config-if)# vrf forwarding v1
Device(config-if)# ipv6 router isis v1b
Device(config-if)# exit
Device(config)# router isis v1a
Device(config-router)# vrf v1
Device(config-router)# net 49.000b.0000.0001.0002.00
Device(config-router)# is-type level-1
Device(config-router)# log-adjacency-changes
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute isis v1b route-map abc
Device(config-router-af)# redistribute isis level-1 into level-2 distribute-list xyz
Device(config-router-af)# exit
Device(config-router)# exit
Device(config)# router isis v1b
Device(config-router)# vrf v1
Device(config-router)# net 49.000b.0000.000a.0001.00
Device(config-router)# log-adjacency-changes
Device(config-router)# address-family ipv6
Device(config-router-af)# redistribute isis v1a route-map abc
Device(config-router-af)# exit
Device(config-router)# exit
Device(config)# route-map abc permit 10
Device(config-route-map)# match route-type level-1
Device(config-route-map)# set metric 56
Device(config-route-map)# set level level-2
```

Additional References for IS-IS IPv6 Multi-Process Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for IS-IS IPv6 Multi-Process Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 175

ISIS Local Microloop Protection

The ISIS Local Microloop Protection feature enables link-state routing protocols, such as the Intermediate System-to-Intermediate System (ISIS) protocol, to prevent or avoid local microloops during network convergence after a link-down event.

Information About ISIS Local Microloop Protection

-

Microloops

When changes occur in a network topology because of the failure or restoration of a link or a network device, IP Fast Reroute enables rapid network convergence by moving traffic to precomputed backup paths until regular convergence mechanisms move traffic to a newly computed best path, also known as a post-convergence path. This network convergence may cause short microloops between two directly or indirectly connected devices in the topology. Microloops are caused when different nodes in the network calculate alternate paths at different times and independently of each other. For instance, if a node converges and sends traffic to a neighbor node, which has not converged yet, traffic may loop between the two nodes.

Microloops may or may not result in traffic loss. If the duration of a microloop is short, that is the network converges quickly, packets may loop for a short duration before their TTL expires. Eventually, the packets will get forwarded to the destination. If the duration of the microloop is long, that is one of the routers in the network is slow to converge, packets may expire their TTL or the packet rate may exceed the bandwidth, and packets may get dropped.

Microloops that are formed between a failed device and its neighbors are called local uloops, whereas microloops that are formed between devices that are multiple hops away are called remote uloops. The ISIS Local Microloop Protection feature helps networks avoid local uloops. Local uloops are usually seen when there is no local loop-free alternate (LFA) path available, especially in ring or square topologies. In such topologies, remote LFAs provide backup paths for the network. However, the fast-convergence benefit of the remote LFA is at risk because of the high probability of uloop creation. The ISIS Local Microloop Protection feature can be used to avoid microloops or local uloops in such topologies.

When to Use Microloop Avoidance

The ISIS Local Microloop Protection feature supports the following local link down events

- Interface-down events
- Adjacency-down events due to BFD sessions going down.
- Adjacency-down events due to neighbor holdtime expiration

The ISIS Local Microloop Protection feature can be used whether or not a topology is supported by loop-free alternates (LFAs). When you use this feature for prefixes that have repair paths installed in the forwarding plane, this feature will support interface-down events and adjacency-down events if bidirectional forwarding detection (BFD) sessions are down. If this feature is used whether or not a repair path has been installed in the forwarding plane, this feature will also support adjacency-down events caused by neighbor holdtime expiration.

The value of using this feature also depends on whether the remote event that caused loss of adjacency on the neighbor is detectable by the local forwarding plane; that is whether the forwarding plane will react and switch to using preprogrammed repair paths. For instance, when a link fails, the reaction time of the local forwarding plane depends on the media. If the media is optical, the failure is likely to be detected within milliseconds, in which case microloop avoidance is useful. If the media is copper, the local detection will be much slower or nonexistent, in which case using microloop avoidance is disadvantageous. However, if the timeout of the neighbor adjacency is due to reasons other than link failure, such as local congestion, lack of CPU time, and long input queues, these reasons are undetectable by the local forwarding plane and therefore, are not good candidates for microloop avoidance.

Note When remote loop-free alternates (RLFAs) are enabled in a network, microloop avoidance is enabled by default for all protected prefixes (prefixes that have repair paths).

How to Configure ISIS Local Microloop Protection

Configuring Microloop Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis**[*area-tag*]
4. **microloop avoidance [disable | protected]**
5. **end**
6. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis**[*area-tag*] **Example:** `Device(config)# router isis`	Enables Intermediate System-to-Intermediate System (IS-IS) as the IP routing protocol and enters router configuration mode.
Step 4	**microloop avoidance [disable | protected]** **Example:** `Device(config-router)# microloop avoidance protected`	Enables local microloop avoidance for protected prefixes. **Note** If you use the **microloop avoidance** command without any of the keywords, microloop avoidance is configured for all prefixes in the network, whether or not they are protected. The protected keyword ensures that microloop avoidance is enabled only for protected prefixes.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Displays the current running configuration.

Modifying the RIB-update value

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **microloop avoidance**[**rib-update-delay** *delay-time*]
5. **end**
6. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Enables Intermediate System-to-Intermediate System (IS-IS) as the IP routing protocol and enters router configuration mode.
Step 4	**microloop avoidance**[**rib-update-delay** *delay-time*] **Example:** `Device(config-router)# microloop avoidance rib-update-delay 6000`	Configures Routing Information Base (RIB) update delay value to avoid microloops in a network.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Displays the current running configuration.

Configuration Examples for ISIS Local Microloop Protection

Example: Configuring Microloop Protection

The following example shows how to configure microloop protection for protected prefixes:

```
Device> enable
Device# configure terminal
Device(config)# router isis
Device(config-router)# microloop avoidance protected
Device(config-router)# end
```

The following example shows how to configure microloop avoidance for protected and unprotected prefixes:

```
Device> enable
Device# configure terminal
Device(config)# router isis
```

```
Device(config-router)# microloop avoidance
Device(config-router)# end
```

The following example shows how to modify the rib-update delay:

```
Device> enable
Device# configure terminal
Device(config)# router isis
Device(config-router)# microloop avoidance rib-update-delay 6000
Device(config-router)# end
```

Additional References for IS-IS Local Microloop Protection

Related Documents

Related Topic	Document Title
IS-IS commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: ISIS Command Reference*
Overview of IS-IS concepts	"Integrated IS-IS Routing Protocol Overview" module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for ISIS Local Microloop Protection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 176

IS-IS Multi-Part TLVs

A TLV is a tuple of (Type, Length, Value) that is used by the IS-IS protocol to advertise information. TLVs contain information called keys which indicate the identity of the object to which the contents of the TLV apply.

In IS-IS, the length of a TLV is limited to 255 bytes. However, based on the local configuration, there could be more than 255 bytes of information to advertise about a particular object. In such cases, multiple TLVs are required. This is referred to as multi-part TLV (MP-TLV).

Currently, Cisco's IS-IS implementation supports the sending of MP-TLVs for prefix-reachability, router capability, and IS-neighbor advertisements, by default. However, in a network, not all routers might support the processing of MP-TLVs. This could result in interoperability problems, including incorrect routing and forwarding.

From Cisco IOS XE 17.13.1 release, you can choose to selectively disable the sending of MP-TLVs based on type. When you disable the sending of MP-TLV, the amount of information sent for a given object is limited to 255 bytes. If the configuration requires sending more than 255 bytes, some information will not be advertised, which could result in incorrect routing.

Warning If a configuration requires sending MP-TLVs and not all routers in the network support MP-TLVs for an object, the operation of the network will be compromised.

Disabling Multi-Part TLVs

To disable multi-part TLV, run the **multi-part tlv disable** command. This command allows you to disable MP-TLV for prefix-reachability, neighbor or router-capability TLVs.

```
Router(config-router)# multi-part-tlv disable ?
 level-1               Disable multi-part tlv in level-1
 level-2               Disable multi-part tlv in level-2
 neighbor              Disable multi-part-tlv for neighbor
 prefix                Disable multi-part tlv for prefix
 router-capability     Disable multi-part tlv for router-capability
  <cr>                <cr>
```

The following is a sample of the disable command executed in a Cisco ASR1000 router running on Cisco IOS XE 17.13.1 image.

```
ASR1k(config-router)# multi-part-tlv disable neighbor
```

Note To enable this feature again, use the **no** form of the **multi-part tlv disable** command.

Verifying Successful Disabling of Multi-Part TLVs

To verify whether the MP-TLV functionality is disabled in your router, run the **show isis** command.

```
Router# show isis protocol
Tag 1:
IS-IS Router: 1 (0x10000)
  System Id: 1720.1600.1001.00  IS-Type: level-1  lsp-mtu: 512
  Manual area address(es):
 49.1234
  Routing for area address(es):
 49.1234
  Interfaces supported by IS-IS:
 Ethernet1/0 - IP - IPv6
 Ethernet0/2 - IP - IPv6
 Ethernet0/1 - IP - IPv6
 Ethernet0/0 - IP - IPv6
 Loopback0 - IP - IPv6
  Redistribute(CLNS):
    static (on by default)
  Distance for L2 CLNS routes: 110
  RRR level: level-1-2
  Generate narrow metrics: none
  Accept narrow metrics:   none
  Generate wide metrics:   level-1-2
  Accept wide metrics:     level-1-2
  Parallel flooding:       suppressed
  Adjacency stagger:       disabled
  IP/IPv6 Redistribution Limit: 10000(Default) Threshold: 75%
    L1 Redistributed routes: 0
  Maintenance Mode ID:     140331958078648
  Maintenance Mode:        disabled
  Maintenance Mode Timer:  stopped (0)
  Graceful Reload state:   GR_NONE
  Multi-part-tlv Disabled: Level-1
    prefix
  Multi-part-tlv Disabled: Level-2
    neighbor prefix router-capability
```

Note the disabled status of Multi-part TLV in the above-mentioned sample show output.

Note When you disable MP-TLV and the local configuration requires sending more than 255 bytes about an object, you will see one of the following in the logs:

- MP-TLVs are disabled, and neighbor TLV <name> in level <num> is advertising partial link-state information.
- MP-TLVs are disabled, and prefix TLV <address> in level <num> is advertising partial link-state information.
- MP-TLVs are disabled, and router-cap TLV advertisement is truncated to fit in one TLV.

PART VI

LISP

CHAPTER 177

Locator ID Separation Protocol (LISP) Overview

Locator ID Separation Protocol (LISP) is a network architecture and protocol that implements the use of two namespaces instead of a single IP address:

- Endpoint identifiers (EIDs)—assigned to end hosts.
- Routing locators (RLOCs)—assigned to devices (primarily routers) that make up the global routing system.

Splitting EID and RLOC functions yields several advantages including improved routing system scalability, and improved multihoming efficiency and ingress traffic engineering.

LISP functionality requires LISP-specific configuration of one or more LISP-related devices, such as the LISP egress tunnel router (ETR), ingress tunnel router (ITR), proxy ETR (PETR), proxy ITR (PITR), map resolver (MR), map server (MS), and LISP alternative logical topology (ALT) device.

Prerequisites for Configuring LISP

Before you can configure Locator/ID Separation Protocol (LISP), you will need to determine the type of LISP deployment you intend to deploy. The LISP deployment defines the necessary functionality of LISP devices, which, in turn, determines the hardware, software, and additional support from LISP mapping services and proxy services that are required to complete the deployment.

LISP configuration requires the datak9 license.

Restrictions for Configuring LISP

- LISP is not supported on Tunnels.
- Management traffic generated on a LISP xTR with the source of a LISP EID interface does not work because management traffic such as SSH or telnet are not LISP aware. To make management protocols LISP aware, you need to create a static route pointing towards correct next hop. The static route should have a next hop of the LISP virtual interface and IP of the RLOC for the remote ETR. Management

traffic generated on an xTR from a LISP EID interface needs this route inserted in the routing table as a workaround of this limitation.

Information About Configuring LISP

LISP Functionality Overview

Problem

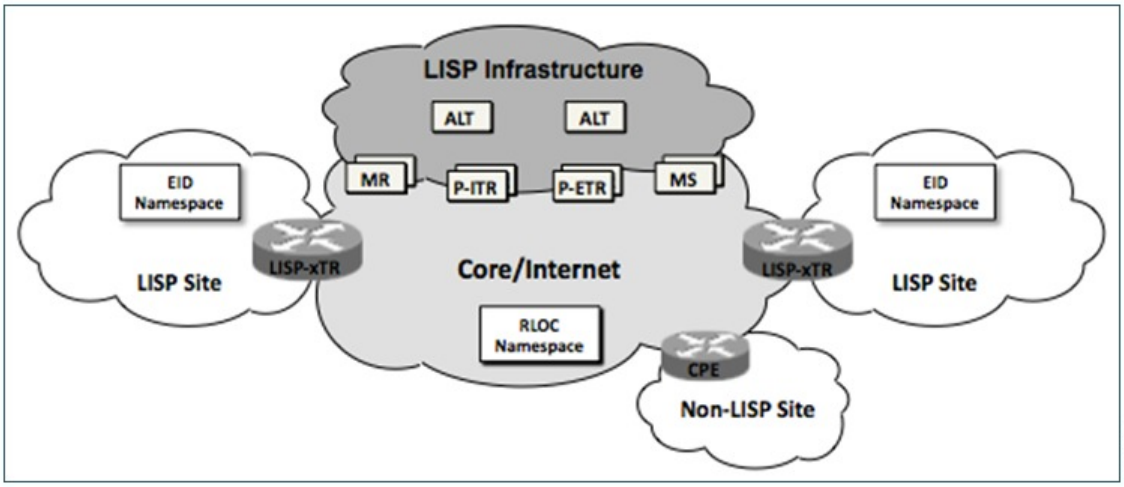
The continuous growth of the Internet presents a number of challenges. Among the most fundamental of these challenges is ensuring that the routing and addressing system continues to function efficiently even as the number of connected devices continues to increase. A basic observation during early network research and development work was that the single IP address, which includes both identity and location, leads to suboptimal route scaling and hinders multihoming and device mobility.

Solution

Locator ID Separation Protocol (LISP) provides improved routing scalability and facilitates flexible address assignment for multi-homing, provider independence, mobility, and virtualization. LISP offers an alternative to traditional Internet architecture by introducing two separate IP addresses: one to indicate routing locators (RLOCs) for routing traffic through the global Internet and a second address for endpoint identifiers (EIDs) used to identify network sessions between devices.

The figure below displays a general overview illustration of a Cisco IOS XE LISP deployment environment, including the three essential environments that exist in a LISP environment: LISP sites (EID namespace), non-LISP sites (RLOC namespace), and LISP mapping service (infrastructure).

Figure 148: Cisco IOS XE LISP Deployment Environment



As illustrated in the figure, the LISP EID namespace represents customer end sites in the same way that end sites are defined in non-LISP environments with one difference: The IP addresses used within these LISP sites are not advertised within the non-LISP Internet (RLOC namespace). Instead, end-customer LISP functionality is deployed exclusively on customer endpoint routers, which perform both the egress tunnel router (ETR) and ingress tunnel router (ITR) functions of a LISP device (abbreviated as xTR in the figure).

To fully implement LISP with support for mapping services and Internet interworking may require additional LISP infrastructure components as part of the deployment. As displayed in the figure above, these additional LISP infrastructure components include devices that function in the LISP roles of map resolver (MR), map server (MS), proxy egress tunnel router (PETR), proxy ingress tunnel router (PITR), and LISP alternative logical topology (ALT) device.

LISP Network Element Functions

The LISP architecture defines seven LISP-specific network infrastructure components. In some cases, a single physical device can implement more than one of these logical components. For more information, refer to the descriptions of the LISP components described in the following sections:

LISP Alternative Logical Topology

An alternative logical topology (ALT) device (not present in all mapping database deployments) connects through generic routing encapsulation (GRE) tunnels and border gateway protocol (BGP) sessions, map resolvers, map servers, and other ALT routers. The only purpose of ALT routers is to accept EID (Endpoint IDentifier) prefixes advertised by devices that form a hierarchically distinct part of the EID numbering space and then advertise an aggregated EID prefix that represents that distinct space to other parts of the ALT. Just as in the global Internet routing system, this aggregation is performed to reduce the number of prefixes that need to be propagated throughout the entire network. An MS or combined MR/MS may also be configured to perform the functions of an ALT router.

LISP Egress Tunnel Router

An ETR connects a site to the LISP-capable part of a core network (such as the Internet), publishes EID-to-RLOC mappings for the site, responds to Map-Request messages, and decapsulates and delivers LISP-encapsulated user data to end systems at the site. During operation, an ETR sends periodic Map-Register messages to all its configured map servers. The Map-Register messages contain all the EID-to-RLOC entries for the EID-numbered networks that are connected to the ETR’s site.

An ETR that receives a Map-Request message verifies that the request matches an EID for which it is authoritative, constructs an appropriate Map-Reply message containing its configured mapping information, and sends this message to the ingress tunnel router (ITR) whose RLOCs are listed in the Map-Request message. An ETR that receives a LISP-encapsulated packet that is directed to one of its RLOCs decapsulates the packet, verifies that the inner header is destined for an EID-numbered end system at its site, and then forwards the packet to the end system using site-internal routing.

The ETR function is usually implemented in the customer premises equipment (CPE) router and does not require hardware changes on software-switched platforms, such as a Cisco Integrated Services Router (ISR). The same CPE router will often provide both ITR and ETR functions and, when doing so, is referred to as an xTR.

LISP Ingress Tunnel Router (ITR)

An ITR is responsible for finding EID-to-RLOC mappings for all traffic destined for LISP-capable sites. When the ITR receives a packet destined for an EID, it first looks for the EID in its mapping cache. If the ITR finds a match, it encapsulates the packet inside a LISP header with one of its RLOCs as the IP source address and one of the RLOCs from the mapping cache entry as the IP destination. The ITR then routes the packet normally.

If no entry is found in the ITR's mapping cache, the ITR sends a Map-Request message to one of its configured map resolvers and then discards the original packet. When the ITR receives a response to its Map-Request

message, it creates a new mapping cache entry with the contents of the Map-Reply message. When another packet, such as a retransmission for the original and, now, discarded packet arrives, the new mapping cache entry is used for encapsulation and forwarding.

Note Sometimes the Map-Reply message will indicate that the destination is not an EID. When this happens, a negative mapping cache entry is created, which causes packets to either be discarded or forwarded natively when the packets match that cache entry.

Like the ETR, an ITR is usually implemented in a LISP site's customer premises equipment (CPE) router, which is typically configured as an xTR (performs functions of both ETR and ITR components).

LISP Map Resolver

Like an MS, a LISP MR connects to the ALT. The function of the LISP MR is to accept encapsulated Map-Request messages from ingress tunnel routers (ITRs), decapsulate those messages, and then forward the messages to the MS responsible for the egress tunnel routers (ETRs) that are authoritative for the requested EIDs.

When an MR is implemented concurrently with an MS in a private mapping system deployment, the concurrent MS forwards the encapsulated Map-Request messages to the authoritative ETRs. When a LISP ALT is present in the deployment, the MR forwards the Map-Request messages directly over the ALT to the MS responsible for the ETRs that are authoritative for the requested EIDs. An MR also sends Negative Map-Replies to ITRs in response to queries for non-LISP addresses.

LISP Map Server

An MS implements part of the distributed LISP mapping database by accepting registration requests from its client egress tunnel routers (ETRs), aggregating the successfully registered EID prefixes of those ETRs, and advertising the aggregated prefixes into the alternative logical topology (ALT) with border gateway protocol (BGP).

In a small private mapping system deployment, an MS may be configured to stand alone (or there may be several MSs) with all ETRs configured to register to each MS. If more than one, all MSs have full knowledge of the mapping system in a private deployment.

In a larger or public mapping system deployment, an MS is configured with a partial mesh of generic routing encapsulation (GRE) tunnels and BGP sessions to other map server systems or ALT routers. For these deployments, ETRs need to register to only one MS (or a few if redundancy is desired) and an ALT device is used to ensure that the entire LISP mapping system is available to all MS and MR devices.

Because an MS does not forward user data traffic—it handles only LISP control plane traffic—it does not require high performance switching capability and is well suited for implementation on a general purpose router, such as a Cisco Integrated Services Router (ISR). Both MS and MR functions are typically implemented on the same device, which is referred to as an MR/MS device.

LISP Proxy ETR

A LISP PETR implements ETR functions on behalf of non-LISP sites. A PETR is typically used when a LISP site needs to send traffic to non-LISP sites but the LISP site is connected through an access network of a service provider that does not accept nonroutable EIDs as packet sources.

When dual-stacked, a PETR may also serve as a way for EIDs and RLOCs to communicate in a LISP site that contains EIDs in one address family and RLOCs in a different address family. A dual-stacked PETR also provides multiaddress family support for LISP EIDs within one address family to be able to communicate with non-LISP destinations in the same address family over a core network within a different address family.

Example

A LISP site with IPv4-only RLOC connectivity can send IPv6 EIDs within an IPv4 LISP header across the IPv4 Internet to a dual-stacked PETR where the packets are decapsulated and then forwarded natively to non-LISP IPv6 Internet sites.

The PETR function is commonly configured on a device that also functions as a PITR. A device that functions as both a PETR and a PITR is known as a PxTR. Additionally, a PETR carries LISP data plane traffic and can be a high packet-rate device. To take advantage of this high packet-rate capability, deployments typically include hardware-switched platforms or high-end Cisco Integrated Services Routers (ISRs).

LISP Proxy ITR

A LISP PITR implements ITR mapping database lookups and LISP encapsulation functions on behalf of non-LISP-capable sites. PITRs are typically deployed near major Internet exchange points (IXPs) or in ISP networks to allow non-LISP customers from those networks to connect to LISP sites. In addition to implementing ITR functionality, a PITR also advertises some or all of the non-routable EID prefix space to the part of the non-LISP-capable Internet that it serves so that the non-LISP sites will route traffic toward the PITR for encapsulation and forwarding to LISP sites.

Note PITR advertising of nonroutable EID prefix space is intended to be highly aggregated with many EID prefixes represented by each prefix that is advertised by a PITR.

Like the PETR, when dual-stacked, the PITR also provides multiple-address family support. But the PITR supports transport of non-LISP traffic from one address family to LISP sites in the same address family over a core network within a different address family.

Example

A LISP site with IPv4-only RLOC connectivity can take advantage of a dual-stacked PITR to allow non-LISP IPv6 Internet users to reach IPv6 EIDs across the IPv4 Internet.

The PITR function is commonly configured on a device that also functions as a PETR. A device that functions as both a PETR and a PITR is known as a PxTR. Additionally, a PITR carries LISP data plane traffic and can be a high packet-rate device. To take advantage of this high packet-rate capability, deployments typically include hardware-switched platforms or high-end Cisco® Integrated Services Routers (ISRs).

Feature Information for LISP Overview

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 191: Feature Information for LISP Overview

Feature Name	Releases	Feature Information
LISP Overview	15.1(4)M Cisco IOS XE Release 3.3.0S	The LISP Overview feature provides a general overview of LISP and its components. The following LISP components are supported: • Egress tunnel router (ETR) • Ingress tunnel router (ITR) • LISP alternative logical topology (ALT) device • Map resolver (MR) • Map server (MS) • Proxy ETR (PETR) • Proxy ITR (PITR)
LISP, SHA-2 support for site registration	15.3(2)T Cisco IOS XE Release 3.9S	LISP can be configured to use SHA2-based HMAC algorithm for integrity-checking LISP site registration messages. Prior to this release, only SHA1-based HMAC algorithm was supported. The following commands were modified: • ipv4 etr map-server • ipv6 etr map-server

CHAPTER 178

Configuring LISP (Locator ID Separation Protocol)

This guide describes how to configure basic Locator ID Separation Protocol (LISP) functionality on all LISP-related devices, including the egress tunnel router (ETR), ingress tunnel router (ITR), proxy ETR (PETR), proxy ITR (PITR), map resolver (MR), and map server (MS).

LISP is a network architecture and protocol that implements the use of two namespaces instead of a single IP address. These namespaces, known as endpoint identifiers (EIDs), are assigned to end-hosts and routing locators (RLOCs), which are assigned to devices (primarily routers) that make up the global routing system. Splitting EID and RLOC functions delivers improvements in routing system scalability, multi-homing efficiency, and ingress traffic engineering.

Prerequisites for Configuring LISP

- If a LISP xTR is also a First Hop Router (FH) or a Rendezvous Point (RP), then the xTR needs to have at least one connected interface that is covered by a local LISP database mapping. Before an ITR forwards traffic over LISP, it does a source check to ensure that the source address of the traffic stream is a local EID (database mapping). Since PIM register and register-stop messages are sourced directly from the router itself, to be forwarded over LISP, the messages must come from an interface covered by a database mapping. A loopback or other connected interface is fine for this purpose. No additional configuration is required to ensure the proper address is selected.

  This prerequisite is not required on a Proxy xTR, which does not do a source check.

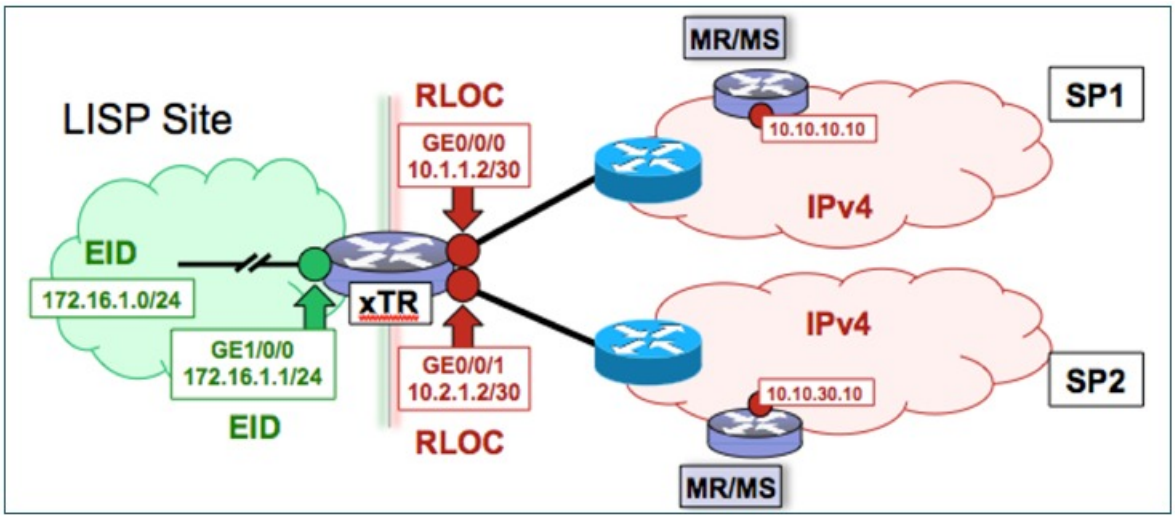
How to Configure LISP

Configure a Dual-Homed LISP Site with Two IPv4 RLOCs and an IPv4 EID

Perform this task to configure a dual-homed LISP site with two IPv4 RLOCs and an IPv4 EID. In this task, a LISP site uses a single edge router configured as both an ITR and an ETR (known as an xTR) with two connections to upstream providers. Both of the RLOCs and the EID prefix are IPv4. The LISP site registers

to two map resolver/map server (MR/MS) devices in the network core. The topology used in this LISP configuration is shown in the figure below.

Figure 149: Dual-Homed LISP Site with Two IPv4 RLOCs and an IPv4 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
  - The CPE functions as a LISP ITR and ETR (xTR).
  - The LISP xTR is authoritative for the IPv4 EID prefix of 172.16.1.0/24.
  - The LISP xTR has two RLOC connections to the core. The RLOC connection to SP1 is 10.1.1.2/30; the RLOC connection to SP2 is 10.2.1.2/30.
  - For this simple dual-homed configuration, the LISP site policy specifies equal load sharing between service provider (SP) links for ingress traffic engineering.
- **Mapping system:**
  - Two map resolver/map server (MR/MS) systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10.
  - Mapping Services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of this LISP site xTR, there is no difference.

Note Map server and map resolver configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map server and map resolver configuration.

This task shows how to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map server and map resolver for mapping services.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. Do one of the following:
   - **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
   - **database-mapping** *EID-prefix/prefix-length* **ipv4-interface** *locator* **priority** *priority* **weight** *weight*
4. Repeat one of the choices in Step 3 to configure a second RLOC.
5. **ipv4 itr**
6. **ipv4 etr**
7. **ipv4 itr map-resolver** *map-resolver-address*
8. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
9. **exit**
10. **ip route** *ipv4-prefix next-hop*
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (Cisco IOS XE software only).

Step 3	Do one of the following:
	• **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
	• **database-mapping** *EID-prefix/prefix-length* **ipv4-interface** *locator* **priority** *priority* **weight** *weight*
	Example:
	`Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`
	Example:
	`Router(config-router-lisp)# database-mapping 172.16.1.0/24 ipv4-interface GigabitEthernet0/0/0 priority 1 weight 50`
		Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
		• In this step example, a single EID prefix, 172.16.1.0/24, is being associated with the single IPv4 RLOC 10.1.1.2 but the *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step.
		• In the second example, the configuration shows the use of the dynamic interface form of the **database-mapping** command. This form is useful when the RLOC address is obtained dynamically, such as via DHCP.

Step 4	Repeat one of the choices in Step 3 to configure a second RLOC.	—

	Command or Action	Purpose
Step 5	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 6	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 7	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	Configures the locator address of the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 8	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures the locator address of the LISP map server and the authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 9	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

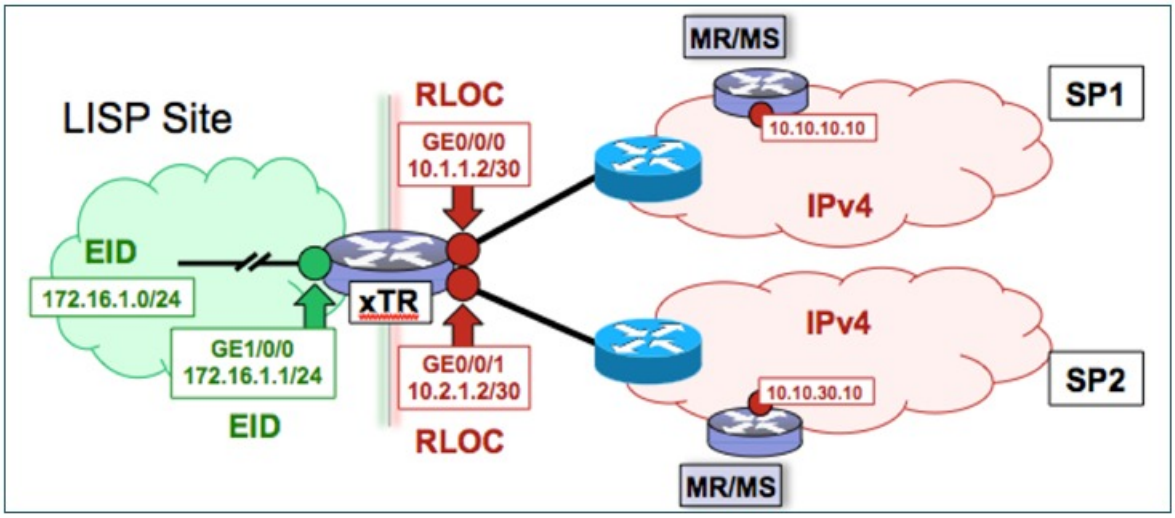
	Command or Action	Purpose

Step 10	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP.
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

Step 11	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 150: Dual-Homed LISP Site with Two IPv4 RLOCs and an IPv4 EID



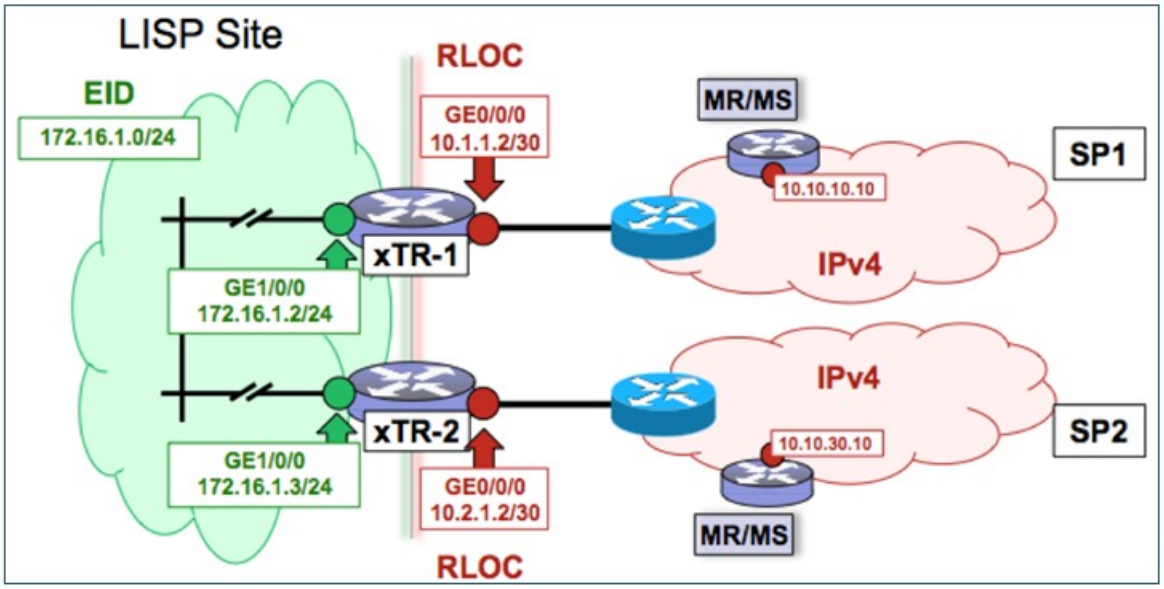
This example shows the complete configuration for the LISP topology illustrated in the figure above and in this task.

```
hostname xTR
!
no ip domain lookup
ip cef
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
!
interface GigabitEthernet0/0/1
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.1 255.255.255.0
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
ip route 0.0.0.0 0.0.0.0 10.2.1.1
```

Configure a Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and an IPv4 EID

Perform this task to configure a multihomed LISP site with two xTRs, two IPv4 RLOCs, and an IPv4 EID. In this task, a LISP site uses two edge routers. Each edge router is configured as an xTR (each performs as both an ITR and an ETR) and each also includes a single IPv4 connection to an upstream provider. (Two different providers are used in this example but the same upstream provider could be used for both connections.) Both of the RLOCs and the EID prefix are IPv4. The LISP site registers to two map resolver/map server (MR/MS) devices in the network core. The topology used in this typical multihomed LISP configuration is shown in the figure below.

Figure 151: Typical Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and an IPv4 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
  - Two CPE routers make up the LISP site: xTR-1 and xTR-2.
  - Both CPE routers function as LISP xTRs (that is, an ITR and an ETR).
  - The LISP site is authoritative for the IPv4 EID prefix of 172.16.1.0/24.
  - Each LISP xTR has a single IPv4 RLOC connection to the core: the RLOC connection for xTR-1 to SP1 is 10.1.1.2/30; the RLOC connection for xTR-2 to SP2 is 10.2.1.2/30.
  - For this multihomed case, the LISP site policy specifies equal load-sharing between service provider (SP) links for ingress traffic engineering.
- **Mapping system:**
  - Two map resolver/map server (MR/MS) systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10.
  - Mapping services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.

Note Map server and map resolver configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map server and map resolver configuration.

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map server and map resolver for mapping services. The

example configurations at the end of this task show the full configuration for configuring two xTRs (xTR1 and xTR2).

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
4. Repeat Step 3 to configure a second RLOC for the same xTR.
5. **ipv4 itr**
6. **ipv4 etr**
7. **ipv4 itr map-resolver** *map-resolver-address*
8. Repeat Step 7 to configure a second locator address for the map resolver.
9. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
10. Repeat Step 9 to configure a second locator address for the map server.
11. **exit**
12. **ip route** *ipv4-prefix next-hop*
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 3	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this step example, a single EID prefix, 172.16.1.0/24, is being associated with a LISP site that contains two separate xTRs. Each xTR has a single IPv4 RLOC connection to the core. In this example, xTR-1 has an IPv4 RLOC connection to SP1 at 10.1.1.2 but the *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step.

	Command or Action	Purpose
		Note Two **database-mapping** commands are required on each xTR to indicate to the mapping system that this LISP site is reachable via these two IPv4 RLOCs. In this example, one RLOC is local (connected) to one xTR and the other is local (connected) to the other xTR.
Step 4	Repeat Step 3 to configure a second RLOC for the same xTR. **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for an xTR on this LISP site. • In this step example, the second RLOC connection for xTR-1 has an IPv4 RLOC connection to SP2 (10.2.1.2). **Note** When a LISP site contains multiple xTRs, all xTRs must be configured with identical **database-mapping** commands to provide the mapping system with consistent information about EID-to-RLOC mappings.
Step 5	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 6	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 7	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 8	Repeat Step 7 to configure a second locator address for the map resolver. **Example:**	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.

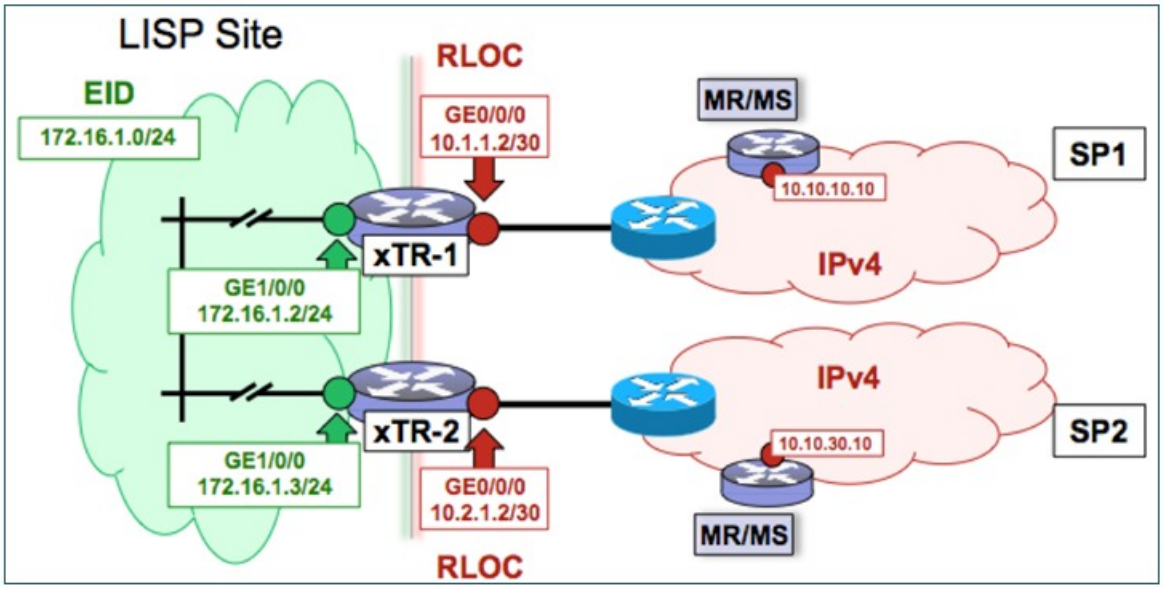
	Command or Action	Purpose
	`Router(config-router-lisp)# ipv4 itr map-resolver 10.10.30.10`	
Step 9	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, each xTR must register to both map servers. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 10	Repeat Step 9 to configure a second locator address for the map server. **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and the authentication key that this router will use to register with the LISP mapping system.
Step 11	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 12	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP

	Command or Action	Purpose
		EID and the destination matches one of the following entries: • a current map-cache entry • a default route with a legitimate next-hop • no route at all In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.
Step 13	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 152: Typical Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and an IPv4 EID



The examples below show the complete configuration for the LISP topology illustrated in the figure above and in this task:

Example configuration for xTR-1:

```
!
hostname xTR-1
!
no ip domain lookup
ip cef
!
```

```
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.2 255.255.255.0
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
```

Example configuration for xTR-2:

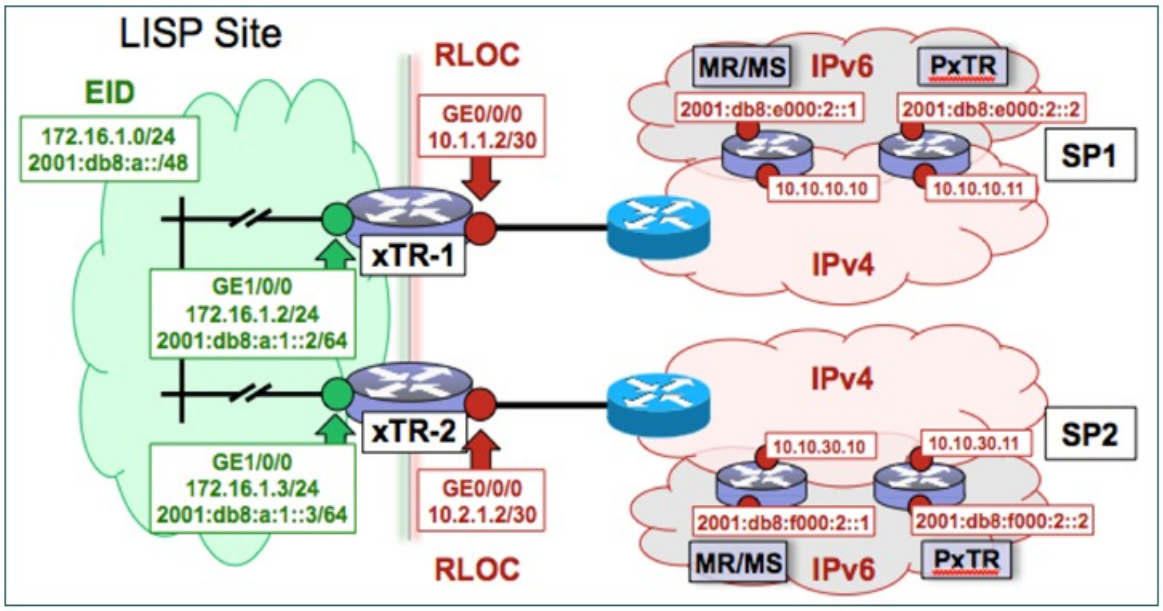
```
!
hostname xTR-2
!
no ip domain lookup
ip cef
!
interface Loopback0
 ip address 172.17.1.2 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.3 255.255.255.0
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.2.1.1
```

Configure a Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and Both an IPv4 and an IPv6 EID

Perform this task to configure a multihomed LISP site with two xTRs, two IPv4 RLOCs, and both an IPv4 and an IPv6 EID. In this task, a LISP site uses two edge routers. Each edge router is configured as an xTR (each performs as both an ITR and an ETR) and each also includes a single IPv4 connection to an upstream provider. (Two different providers are used in this example but the same upstream provider could be used for both connections.) Both of the RLOCs and one of the EIDs are IPv4. However, in this example, the LISP site includes an IPv6 EID, as well.

This LISP site requires the use of Proxy Ingress/Egress Tunnel Router (PxTR) LISP infrastructure for access to non-LISP IPv6 addresses. That is, the LISP site uses only its IPv4 RLOCs to reach IPv6 LISP and non-LISP addresses. Additionally, this LISP site registers to two map resolver/map server (MR/MS) devices in the network core. The topology used in this multihomed LISP configuration is shown in the figure below.

Figure 153: Multihomed LISP Site with Two xTRs, Two IPv4 RLOCs, and Both an IPv4 and an IPv6 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
  - Two CPE routers make up the LISP site: xTR-1 and xTR-2.
  - Both CPE routers function as LISP xTRs (that is, an ITR and an ETR).
  - The LISP site is authoritative for both the IPv4 EID prefix of 172.16.1.0/24 and the IPv6 EID prefix 2001:db8:a::/48.
  - Each LISP xTR has a single RLOC connection to the core: the RLOC connection for xTR-1 to SP1 is 10.1.1.2/30; the RLOC connection for xTR-2 to SP2 is 10.2.1.2/30.
  - For this multihomed case, the LISP site policy specifies equal load-sharing between service provider (SP) links for ingress traffic engineering.

- **Mapping system:**
  - Two map resolver/map server (MR/MS) systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10.
  - Mapping services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.

Note Map server and map resolver configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map server and map resolver configuration.

  - PxTR services are also assumed to be provided as part of this LISP solution via a private or public mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.
  - The PxTRs have IPv4 RLOCs of 10.10.10.11 and 10.10.30.11 and will be used (as PETRs) for LISP IPv6 EIDs to reach non-LISP IPv6 sites. Return traffic is attracted by the PITR function (with the assumption that the PITR advertises coarse aggregates for IPv6 LISP EIDs into the IPv6 core.)

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map server and map resolver for mapping services. The example configurations at the end of this task show the full configuration for two xTRs (xTR1 and xTR2).

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
4. Repeat Step 3 to configure a second RLOC (10.2.1.2) for the same xTR and IPv4 EID prefix.
5. Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix.
6. **ipv4 itr**
7. **ipv4 etr**
8. **ipv4 itr map-resolver** *map-resolver-address*
9. Repeat Step 8 to configure a second locator address of the map resolver.
10. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
11. Repeat Step 10 to configure a second locator address for the map server.
12. **ipv6 itr**
13. **ipv6 etr**
14. **ipv6 itr map-resolver** *map-resolver-address*
15. Repeat Step 14 to configure a second locator address for the map resolver.
16. **ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key*
17. Repeat Step 16 to configure a second locator address for the map server.
18. **ipv6 use-petr** *petr-address*
19. Repeat Step 18 to configure a second locator address for the PETR.

20. **exit**
21. **ip route** *ipv4-prefix next-hop*
22. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 3	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In steps 3, 4, and 5 of this example, an IPv4 EID prefix, 172.16.1.0/24, and an IPv6 prefix, 2001:db8:a::/48, are being associated with a LISP site that contains two separate xTRs that each have a single IPv4 RLOC connection to the core. In this first step example, xTR-1 is configured with an IPv4 RLOC connection to SP1 at 10.1.1.2 but the *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step. **Note** Four **database-mapping** commands are required for each xTR to indicate to the mapping system that both the associated IPv4 and IPv6 EID prefixes are reachable at this LISP site via these two IPv4 RLOCs. In this example, one RLOC is local (connected) to one xTR and the other is local (connected) to the other xTR.
Step 4	Repeat Step 3 to configure a second RLOC (10.2.1.2) for the same xTR and IPv4 EID prefix. **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for an xTR on this LISP site. • In this step example, the second RLOC connection for xTR-1 has an IPv4 RLOC connection to SP2 (10.2.1.2). **Note** When a LISP site contains multiple xTRs, all xTRs must be configured with identical **database-mapping** commands to provide the mapping system with consistent information about EID-to-RLOC mappings.

	Command or Action	Purpose
Step 5	Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix.	—
Step 6	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 7	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 8	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 9	Repeat Step 8 to configure a second locator address of the map resolver. **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.30.10`	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 10	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, each xTR must register to both map servers. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key.

	Command or Action	Purpose
		Note The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 11	Repeat Step 10 to configure a second locator address for the map server. **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and the authentication key that this router will use to register with the LISP mapping system.
Step 12	**ipv6 itr** **Example:** `Router(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 13	**ipv6 etr** **Example:** `Router(config-router-lisp)# ipv6 etr`	Enables LISP ETR functionality for the IPv6 address family.
Step 14	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.10.10`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 15	Repeat Step 14 to configure a second locator address for the map resolver. **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.30.10`	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 16	**ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:**	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system.

	Command or Action	Purpose
	`Router(config-router-lisp)# ipv6 etr map-server 10.10.10.10 key 0 some-key`	• In this example, each xTR must register to both map servers. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 17	Repeat Step 16 to configure a second locator address for the map server. **Example:** `Router(config-router-lisp)# ipv6 itr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register with the LISP mapping system.
Step 18	**ipv6 use-petr** *petr-address* **Example:** `Router(config-router-lisp)# ipv6 use-petr 10.10.10.11`	Configures a locator address for the Proxy Egress Tunnel Router (PETR) to which each xTR will forward LISP-encapsulated IPv6 EIDs (using the xTR's IPv4 RLOC) to reach non-LISP IPv6 addresses. **Note** The PETR is assumed to be dual-stacked and capable of natively reaching the non-LISP IPv6 address. In addition, the PITR is assumed to be dual-stacked and to be advertising coarse aggregates for IPv6 LISP EIDs into the IPv6 core to handle return traffic (non-LISP IPv6 to LISP IPv6 over an IPv4 infrastructure). **Note** The locator address of the PETR may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the PETR is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to eight PETRs may be configured if multiple PETRs are available. (See the *LISP Command Reference* for more details.)
Step 19	Repeat Step 18 to configure a second locator address for the PETR.	Configures a second locator address for the PETR to which each xTR will forward LISP-encapsulated IPv6 EIDs

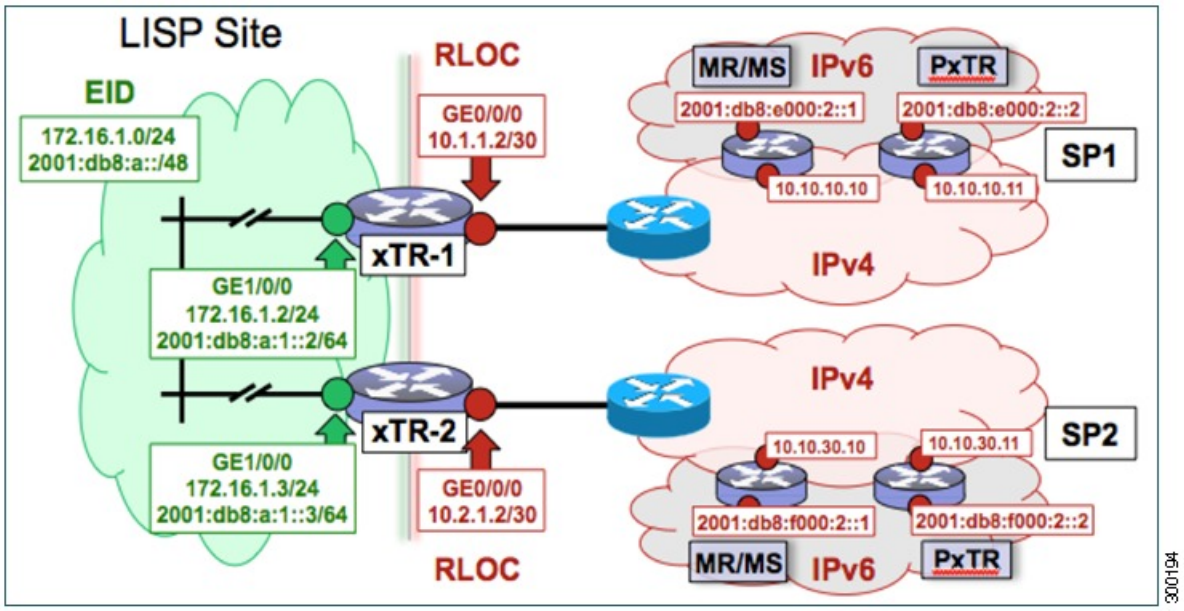
	Command or Action	Purpose
	Example: `Router(config-router-lisp)# ipv6 use-petr 10.10.30.11`	(using the xTR's IPv4 RLOC) to reach non-LISP IPv6 addresses.
Step 20	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 21	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

Step 22	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 154: Multihomed LISP Site with Two xTRs, Two IPv4 RLOCs, and Both an IPv4 and an IPv6 EID



The examples below show the complete configuration for the LISP topology illustrated in the figure above and in this task:

Example configuration for xTR-1:

```
!
hostname xTR-1
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.2 255.255.255.0
 ipv6 address 2001:db8:a:1::2/64
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 ipv4 itr
```

```
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 etr map-server 10.10.10.10 key 0 some-key
 ipv6 etr map-server 10.10.30.10 key 0 some-key
 ipv6 use-petr 10.10.10.11
 ipv6 use-petr 10.10.30.11
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
ipv6 route ::/0
```

Example configuration for xTR-2:

```
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.2 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.3 255.255.255.0
 ipv6 address 2001:db8:a:1::3/64
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv4 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv6 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 use-petr 10.10.10.11
 ipv6 use-petr 10.10.30.11
 exit
!
ip route 0.0.0.0 0.0.0.0 10.2.1.1
```

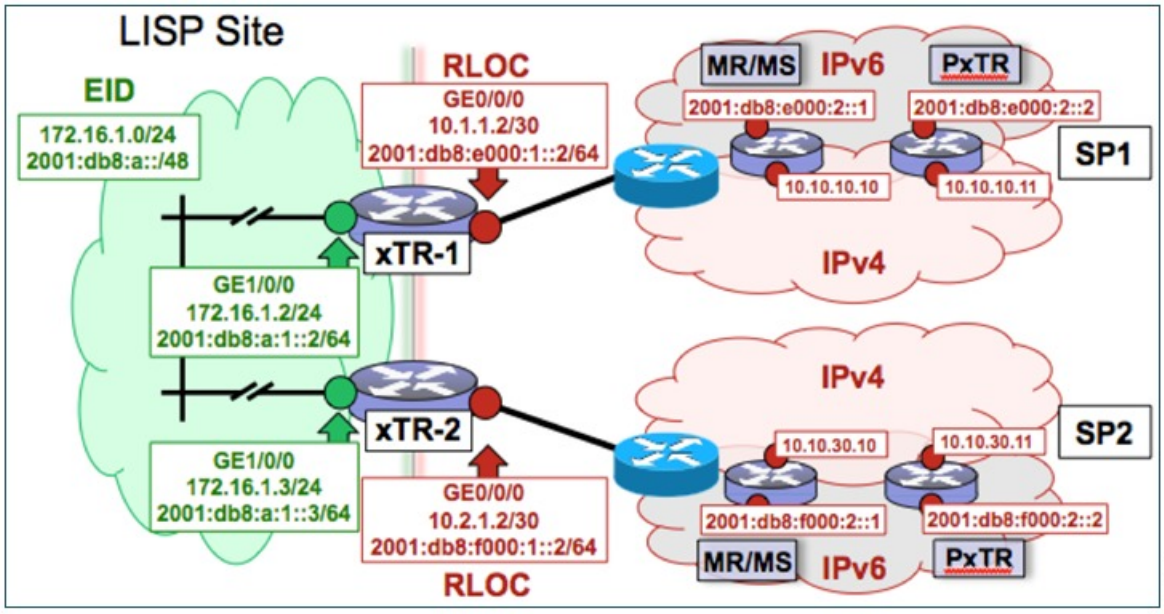
```
!
ipv6 route ::/0
```

Configure a Multihomed LISP Site with Two xTRs that Each have Both an IPv4 and an IPv6 RLOC and Both an IPv4 and an IPv6 EID

Perform this task to configure a multihomed LISP site with two xTRs, each with both an IPv4 and an IPv6 RLOC and both with an IPv4 and an IPv6 EID. In this task, a LISP site uses two edge routers. Each edge router is configured as an xTR (each performs as both an ITR and an ETR) and each also includes a single, dual stack (IPv4 and IPv6) connection to an upstream provider. (Two different providers are used in this example but the same upstream provider could be used for both connections.) Each xTR has an IPv4 RLOC and an IPv6 RLOC and both IPv4 and IPv6 EID prefixes are being used within the LISP site. However, because the site has both IPv4 and IPv6 RLOCs, it does not require a Proxy Ingress/Egress Tunnel Router (PxTR) LISP infrastructure for access to non-LISP IPv6 addresses. (The PxTR infrastructure can still be configured as a resiliency mechanism if desired.)

The LISP site registers to two map resolver/map server (MR/MS) devices in the network core using both IPv4 and IPv6 locators. The topology used in this multihomed LISP configuration is shown in the figure below.

Figure 155: Multihomed LISP Site with Two xTRs, Each with an IPv4 and an IPv6 RLOC and each with an IPv4 and an IPv6 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
  - Two CPE routers make up the LISP site: xTR-1 and xTR-2.
  - Both CPE routers function as LISP xTRs (that is, an ITR and an ETR).
  - The LISP site is authoritative for both the IPv4 EID prefix of 172.16.1.0/24 and the IPv6 EID prefix 2001:db8:a::/48.
  - Each LISP xTR has a single IPv4 RLOC connection and a single IPv6 RLOC connection to the core: the RLOC connections for xTR-1 to SP1 include an IPv4 RLOC, 10.1.1.2/30, and an IPv6

RLOC, 2001:db8:e000:1::2/64. The xTR-2 connections to SP2 include IPv4 RLOC 10.2.1.2/30 and IPv6 RLOC 2001:db8:f000:1::2/64.

- For this multihomed case, the LISP site policy specifies equal load-sharing between service provider (SP) links for ingress traffic engineering.

- **Mapping system:**
  - Two map resolver/map server systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10 and IPv6 RLOCs 2001:db8:e000:2::1 and 2001:db8:f000:2::1.
  - Mapping services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.

Note Map resolver and map server configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map resolver and map server configuration.

  - PxTR services are not required in this example since both xTRs have dual-stack connectivity to the core.

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map resolver and map server for mapping services. The example configurations at the end of this task show the full configuration for two xTRs (xTR1 and xTR2).

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
4. Repeat Step 3 to configure a second IPv4 RLOC for the same xTR and IPv4 EID prefix.
5. Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix.
6. Repeat Step 3, Step 4, and Step 5 to configure the second set of IPv4 and IPv6 RLOC connections on the same xTR for both the IPv4 and IPv6 EID prefixes.
7. **ipv4 itr**
8. **ipv4 etr**
9. **ipv4 itr map-resolver** *map-resolver-address*
10. Repeat Step 9 to configure a second locator address of the LISP map resolver.
11. Repeat Step 9 and Step 10 to configure the IPv6 locator addresses of the LISP two map resolvers.
12. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
13. Repeat Step 12 to configure a second locator address of the map server.
14. Repeat Step 12 and Step 13 to configure the IPv6 locator addresses of the two map servers.
15. **ipv6 itr**
16. **ipv6 etr**
17. **ipv6 itr map-resolver** *map-resolver-address*

18. Repeat Step 17 to configure a second IPv6 locator address of the LISP map resolver.
19. Repeat Step 17 and Step18 to configure the IPv6 (instead of IPv4) locator addresses for the two map resolvers to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions.
20. **ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key*
21. Repeat Step 20 to configure a second locator address of the LISP map server.
22. Repeat Steps 20 and 21 to configure the IPv6 locator addresses of the two map servers for which this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system.
23. **exit**
24. **ip route** *ipv4-prefix next-hop*
25. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 3	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this example, a single IPv4 EID prefix, 172.16.1.0/24, and a single IPv6 prefix, 2001:db8:a::/48, are being associated with a LISP site that contains two separate xTRs that each have a single IPv4 RLOC connection and a single IPv6 connection to the core. In this first database-mapping step example, xTR-1 is configured with an IPv4 RLOC connection to SP1 (10.1.1.2) and an IPv6 RLOC connection to SP1 (2001:db8:e000:1::2/64.) while xTR-2 has an IPv4 RLOC connection of10.2.1.2 to SP2 and an IPv6 RLOC connection of 2001:db8:f000:1::2/64 to SP2. The *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step.

	Command or Action	Purpose
		Note Eight **database-mapping** commands are required for each xTR to indicate to the mapping system that both the IPv4 and IPv6 EID prefixes are reachable at this LISP site via both the two IPv4 RLOCs and the two IPv6 RLOCs. In this example, one IPv4 RLOC and one IPv6 RLOC are local (connected) to one xTR and the others are local (connected) to the other xTR.
Step 4	Repeat Step 3 to configure a second IPv4 RLOC for the same xTR and IPv4 EID prefix. **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for an xTR on this LISP site. • In this step example, the second RLOC connection for xTR-1 has an IPv4 RLOC connection to SP2 (10.2.1.2). **Note** When a LISP site contains multiple xTRs, all xTRs must be configured with identical **database-mapping** commands to provide the mapping system with consistent information about EID-to-RLOC mappings.

Step 5	Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix.
	Example:
	`Router(config-router-lisp)# database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50`
	Example:
	`Router(config-router-lisp)# database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50`
		—

Step 6	Repeat Step 3, Step 4, and Step 5 to configure the second set of IPv4 and IPv6 RLOC connections on the same xTR for both the IPv4 and IPv6 EID prefixes.	—
Step 7	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 8	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.

	Command or Action	Purpose
Step 9	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map resolver is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 10	Repeat Step 9 to configure a second locator address of the LISP map resolver. **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.30.10`	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 11	Repeat Step 9 and Step 10 to configure the IPv6 locator addresses of the LISP two map resolvers.	—
Step 12	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, a second xTR can be registered to the same two map servers using the same authentication key. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map server is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 13	Repeat Step 12 to configure a second locator address of the map server.	Configures a second IPv4 locator address of the LISP map server and the authentication key that this router, acting

	Command or Action	Purpose
	Example: `Router(config-router-lisp)# ipv4 etr map-server 10.10.30.10 key 0 some-key`	as an IPv4 LISP ETR, will use to register with the LISP mapping system.

Step 14	Repeat Step 12 and Step 13 to configure the IPv6 locator addresses of the two map servers.
	Example:
	`ipv4 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key`
	Example:
	`ipv4 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key`
		—

Step 15	**ipv6 itr** **Example:** `Router(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 16	**ipv6 etr** **Example:** `Router(config-router-lisp)# ipv6 etr`	Enables LISP ETR functionality for the IPv6 address family.
Step 17	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.10.10`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map resolver is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 18	Repeat Step 17 to configure a second IPv6 locator address of the LISP map resolver. **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.30.10`	Configures a second locator address of the map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions.
Step 19	Repeat Step 17 and Step18 to configure the IPv6 (instead of IPv4) locator addresses for the two map resolvers to	—

	Command or Action	Purpose

	which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions.
	Example:
	`ipv6 itr map-resolver 2001:db8:e000:2::1`
	Example:
	`ipv6 itr map-resolver 2001:db8:f000:2::1`	

Step 20	**ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system. • In this example, a second xTR can be registered to the same two map servers using the same authentication key. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map server is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 21	Repeat Step 20 to configure a second locator address of the LISP map server. **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register with the LISP mapping system.

Step 22	Repeat Steps 20 and 21 to configure the IPv6 locator addresses of the two map servers for which this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system.
	Example:
	`ipv6 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key`
	Example:
	`ipv6 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key`
		—

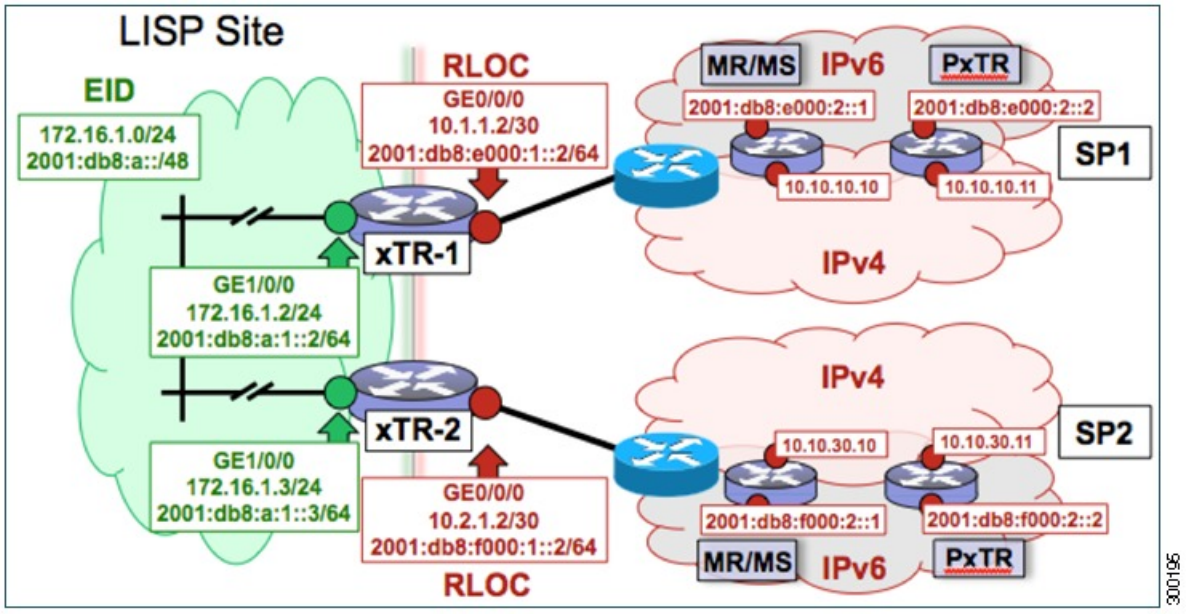
	Command or Action	Purpose
Step 23	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 24	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

Step 25	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 156: Multihomed LISP Site with Two xTRs, Each with an IPv4 and an IPv6 RLOC and each with an IPv4 and an IPv6 EID



The examples below show the complete configuration for the LISP topology illustrated in the figure above and in this task:

Example configuration for xTR-1:

```
!
hostname xTR-1
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
 ipv6 address 2001:db8:e000:1::2/64
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.2 255.255.255.0
 ipv6 address 2001:db8:a:1::2/64
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
```

```
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 2001:db8:e000:1::2 priority 1 weight 50
 database-mapping 172.16.1.0/24 2001:db8:f000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:e000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:f000:1::2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 itr map-resolver 2001:db8:e000:2::1
 ipv4 itr map-resolver 2001:db8:f000:2::1
 ipv4 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv4 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 itr map-resolver 2001:db8:e000:2::1
 ipv6 itr map-resolver 2001:db8:f000:2::1
 ipv6 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv6 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
ipv6 route ::/0 2001:db8:e000:1::1
!
```

Example configuration for xTR-2:

```
!
hostname xTR-2
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.2 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
 ipv6 address 2001:db8:f000:1::2/64
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.3 255.255.255.0
 ipv6 address 2001:db8:a:1::3/64
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 2001:db8:e000:1::2 priority 1 weight 50
```

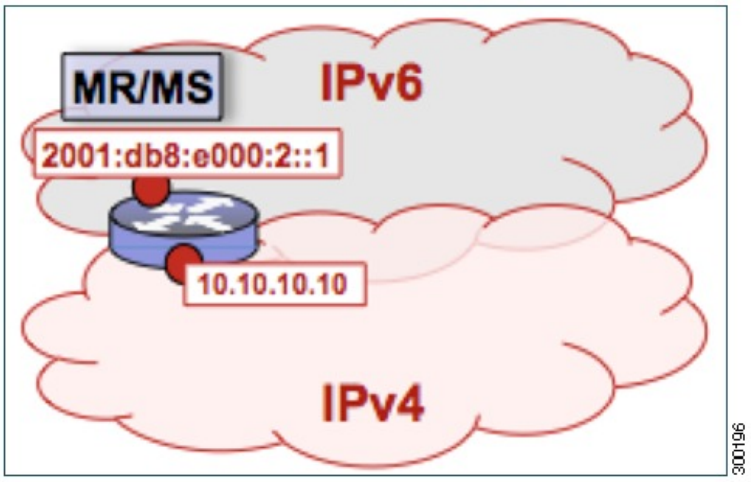
```
 database-mapping 172.16.1.0/24 2001:db8:f000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:e000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:f000:1::2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 itr map-resolver 2001:db8:e000:2::1
 ipv4 itr map-resolver 2001:db8:f000:2::1
 ipv4 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv4 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 itr map-resolver 2001:db8:e000:2::1
 ipv6 itr map-resolver 2001:db8:f000:2::1
 ipv6 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv6 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.2.1.1
!
ipv6 route ::/0 2001:db8:f000:1::1
!
```

Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server

Perform this task to configure and enable standalone LISP map resolver/map server (MR/MS) functionality for both IPv4 and IPv6 address families. In this task, a Cisco device is configured as a standalone MR/MS for a private LISP mapping system. Because the MR/MS is configured as a standalone device, it has no need for LISP alternative logical topology (ALT) connectivity. All relevant LISP sites must be configured to register with this map server so that this map server has full knowledge of all registered EID prefixes within the (assumed) private LISP system. However, because this device is functioning as a map resolver/map server, the data structure associated with an ALT virtual routing and forwarding (VRF) table must still be configured to hold LISP EIDs for registered sites.

The map resolver/map server is configured with both IPv4 and IPv6 RLOC addresses. The topology used in this most basic LISP MR/MS configuration is shown in the figure below.

Figure 157: Standalone LISP Map Resolver/Map Server with both IPv4 and IPv6 RLOCs



The components illustrated in the topology shown in the figure are described below, although the map resolver is configured separately:

Mapping System

- The LISP device is configured to function as a standalone map resolver/map server (MR/MS).
- The xTRs in the LISP site are assumed to be registered to this map server. That is, the xTR registers the IPv4 EID prefix of 172.16.1.0/24 and, when IPv6 EIDs are used, the xTR also registers the IPv6 EID of prefix 2001:db8:a::/48.
- The MR/MS has an IPv4 locator of 10.10.10.10/24 and an IPv6 locator of 2001:db8:e000:2::1/64.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **address-family ipv4** [**unicast**]
4. **exit-address-family**
5. **address-family ipv6**
6. **exit-address-family**
7. **exit**
8. **router lisp**
9. **ipv4 alt-vrf** *vrf-name*
10. **ipv4 map-server**
11. **ipv4 map-resolver**
12. **ipv6 alt-vrf** *vrf-name*
13. **ipv6 map-server**
14. **ipv6 map-resolver**
15. **site** *site-name*
16. **eid-prefix** *EID-prefix*
17. **authentication-key** [*key-type*] *authentication-key*
18. **exit**

19. Repeat Steps 15 through 18 to configure additional LISP sites.
20. **exit**
21. **ip route** *ipv4-prefix next-hop*
22. **ipv6 route** *ipv6-prefix next-hop*
23. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition lisp`	Creates a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF table. In this example, a VRF table named lisp is created to hold EID prefixes.
Step 3	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF table named lisp handles IPv4 EID prefixes.
Step 4	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv4 address family configuration mode and returns to VRF configuration mode.
Step 5	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 6	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv6 address family configuration mode and returns to VRF configuration mode.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

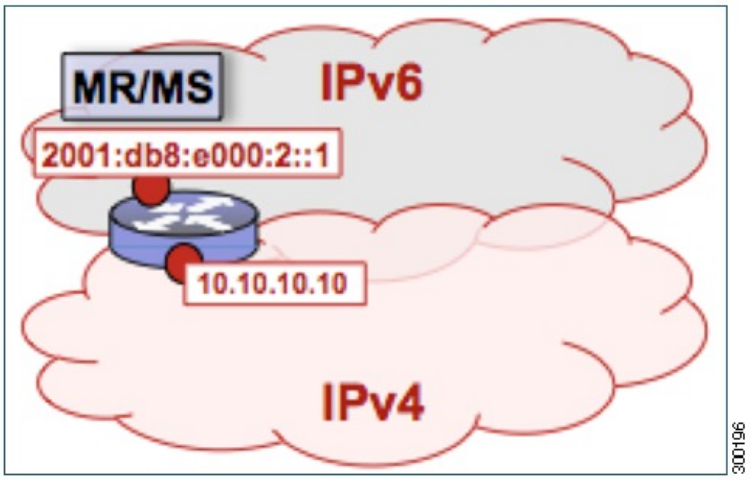
	Command or Action	Purpose
Step 8	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (Cisco IOS XE software only).
Step 9	**ipv4 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv4 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv4 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 10	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 11	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 12	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 13	**ipv6 map-server** **Example:** `Router(config-router-lisp)# ipv6 map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family.
Step 14	**ipv6 map-resolver** **Example:** `Router(config-router-lisp)# ipv6 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family.
Step 15	**site** *site-name* **Example:** `Router(config-router-lisp)# site Site-1`	Specifies a LISP site named Site-1 and enters LISP site configuration mode. **Note** A LISP site name is locally significant to the map server on which it is configured. It has no relevance anywhere else. This name is used solely as an administrative means of associating one or more EID prefixes with an authentication key and other site-related mechanisms.
Step 16	**eid-prefix** *EID-prefix* **Example:**	Configures an IPv4 or IPv6 EID prefix associated with this LISP site.

	Command or Action	Purpose
	`Router(config-router-lisp-site)# eid-prefix 172.16.1.0/24`	• Repeat this step as necessary to configure additional EID prefixes under this LISP sites. • In this step example, only an IPv4 EID prefix is configured but to complete the configuration, an IPv6 EID prefix must also be configured. **Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key. **Note** Additional **eid-prefix** command configuration options are available. (See the *LISP Command Reference* for more details.)
Step 17	**authentication-key** [*key-type*] *authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 some-key`	Configures the authentication key associated with this site. **Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key. **Note** The **authentication-key** can be configured with Type 6 encryption. (See the *LISP Command Reference* for more details.)
Step 18	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 19	Repeat Steps 15 through 18 to configure additional LISP sites.	—
Step 20	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 21	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 22	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:1::1`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 23	**exit** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config)# exit`	

Example:

Figure 158: Standalone LISP Map Resolver/Map Server with both IPv4 and IPv6 RLOCs



The example below shows the complete configuration for the LISP topology illustrated in the figure above and in this task. However, this example is for a full configuration of a standalone LISP MR/MS and includes some basic IPv4 and IPv6 configuration not covered in this task:

```
!
hostname MR-MS
!
vrf definition lisp
!
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.2.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.10.10.10 255.255.255.0
 ipv6 address 2001:db8:e000:2::1/64
!
router lisp
 site Site-1
```

```
  authentication-key some-key
  eid-prefix 172.16.1.0/24
  eid-prefix 2001:db8:a::/48
  exit
 !
 site Site-2
  authentication-key another-key
  eid-prefix 172.16.2.0/24
  eid-prefix 2001:db8:b::/48
  exit
 !
!---more LISP site configs---
 !
 ipv4 map-server
 ipv4 map-resolver
 ipv4 alt-vrf lisp
 ipv6 map-server
 ipv6 map-resolver
 ipv6 alt-vrf lisp
 exit
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
!
ipv6 route ::/0 2001:db8:e000:2::fof
```

Configure a Public Mapping System Using Separate ALT-Connected Map Resolver and Map Server Devices

The following tasks show how to configure a map resolver (MR) and a map server (MS) on separate devices, each using LISP alternative logical topology (ALT) connectivity. The MR and MS share their EID prefix information via the LISP ALT connectivity, which is typical of a public LISP deployment model where higher performance and scalability (for tasks such as the handling of Map-Request messages) is required. The LISP ALT is implemented as an overlay virtualized network using GRE tunnels and BGP, which allows for separation of EID prefixes from the underlying core network.

Configuring an ALT-Connected LISP Map Resolver

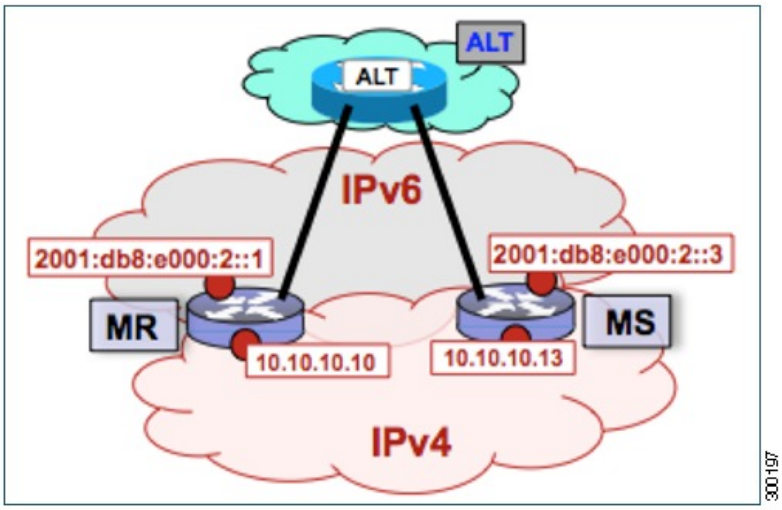
Before you begin

Perform this task to configure LISP alternative logical topology (ALT) map resolver functionality for both IPv4 and IPv6 address family mapping services on Cisco IOS XE Everest 16.6.1 and later releases.

Note You must also configure an ALT-connected LISP map server (see the Configuring an ALT-Connected LISP Map Server task).

In the figure below, the map resolver (MR) and map server (MS) are configured on separate devices and share their EID prefix information via connectivity.

Figure 159: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The map resolver illustrated in the topology shown in the figure is described below; the map server and LISP ALT are configured in separate tasks:

Mapping System

- Two LISP devices are configured, one as an MS and the other as an MR.
- The MS has an IPv4 locator of 10.10.10.13/24 and an IPv6 locator of 2001:db8:e000:2::3/64.
- The MR has an IPv4 locator of 10.10.10.10/24 and an IPv6 locator of 2001:db8:e000:2::1/64.
- Assume that the xTRs in the LISP site register to this map server. That is, the xTR registers the IPv4 EID-prefix of 172.16.1.0/24 and, when IPv6 EIDs are used, the xTR registers the IPv6 EID-prefix of 2001:db8:a::/48.

Note The configuration of the xTR must be changed to use the MS RLOC for its map server configuration and the MR RLOC for its map resolver configuration. For example:

- **ipv4 itr map-resolver 10.10.10.10**
- **ipv4 etr map-server 10.10.10.13 key 0 some-key**

Other Infrastructure

- The MR has IPv4 and IPv6 tunnel endpoints in the VRF table (named lisp) of 192.168.1.1/30 and 2001:db8:ffff::1/64, respectively, and the MS has IPv4 and IPv6 tunnel endpoints of 192.168.1.2/30 and 2001:db8:ffff::2/64, respectively, in the same VRF table. This tunnel is used for the ALT.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family ipv4** [**unicast**]

5. **exit-address-family**
6. **address-family ipv6**
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **ipv6 address** *ipv6-address/mask*
13. **tunnel source** *interface-type interface-number*
14. **tunnel destination** *ipv4-address*
15. **exit**
16. **router lisp**
17. **ipv4 map-resolver**
18. **ipv4 alt-vrf** *vrf-name*
19. **ipv6 map-resolver**
20. **ipv6 alt-vrf** *vrf-name*
21. **exit**
22. **router bgp** *autonomous-system-number*
23. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
24. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
25. **neighbor** *ip-address* **activate**
26. **exit**
27. **address-family ipv6 vrf** *vrf-name*
28. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
29. **neighbor** *ip-address* **activate**
30. **exit**
31. **exit**
32. **ip route** *ipv4-prefix next-hop*
33. **ipv6 route** *ipv6-prefix next-hop*
34. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition lisp`	Creates a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF. In this example, a VRF named lisp is created to hold EID prefixes.

	Command or Action	Purpose
Step 3	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.
Step 4	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF table named lisp handles IPv4 EID prefixes.
Step 5	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv4 address family configuration mode and returns to VRF configuration mode.
Step 6	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv6 address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface tunnel 192`	Specifies the interface type of tunnel and the interface number and enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding lisp`	Associates a VRF instance configured in Step 2 with the tunnel interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.1 255.255.255.252`	Configures an IPv4 address for the tunnel interface.
Step 12	**ipv6 address** *ipv6-address/mask* **Example:**	Configures an IPv6 address for the tunnel interface.

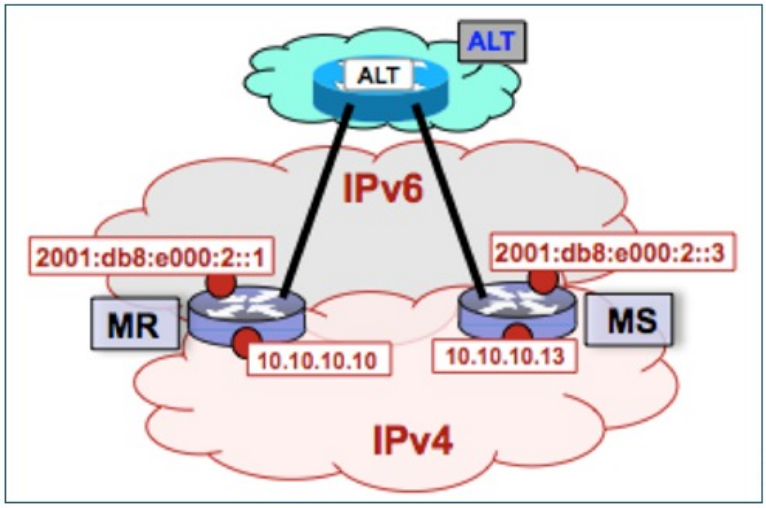
	Command or Action	Purpose
	`Router(config-if)# ipv6 address 2001:db8:ffff::1/64`	
Step 13	**tunnel source** *interface-type interface-number* **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Configures the tunnel source.
Step 14	**tunnel destination** *ipv4-address* **Example:** `Router(config-if)# tunnel destination 10.10.10.13`	Configures the tunnel destination IPv4 address for the tunnel interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 16	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 17	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 18	**ipv4 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv4 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv4 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 19	**ipv6 map-resolver** **Example:** `Router(config-router-lisp)# ipv6 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family.
Step 20	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 21	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 22	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65010`	Enters router configuration mode for the specified routing process.

	Command or Action	Purpose
Step 23	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf lisp`	Specifies the IPv4 address family and enters IPv4 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv4 VRF that carries EID-prefixes in the LISP ALT.
Step 24	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.2 remote-as 65011`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 25	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family.
Step 26	**exit** **Example:** `Router(config-router-af)# exit`	Exits IPv4 address family configuration mode and returns to router configuration mode.
Step 27	**address-family ipv6 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv6 vrf lisp`	Specifies the IPv6 address family and enters IPv6 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv6 VRF that carries EID-prefixes in the LISP ALT.
Step 28	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::2 remote-as 65011`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 29	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::2 activate`	Enables the neighbor to exchange prefixes for the IPv6 unicast address family.

	Command or Action	Purpose
Step 30	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 31	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 32	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 33	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 34	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Examples

Figure 160: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a LISP map resolver including some basic IP and IPv6 configuration not included in the task table for this task:

```
!
```

```
vrf definition lisp
 rd 1:1
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface Tunnel192
 vrf forwarding lisp
 ip address 192.168.1.1 255.255.255.252
 ipv6 address 2001:db8:ffff::1/64
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.10.10.13
!
interface GigabitEthernet 0/0/0
 description Link to SP1 (RLOC)
 ip address 10.10.10.10 255.255.255.0
 ipv6 address 2001:db8:e000:2::1/64
!
router lisp
 ipv4 map-resolver
 ipv4 alt-vrf lisp
 ipv6 map-resolver
 ipv6 alt-vrf lisp
 exit
!
router bgp 65010
 bgp asnotation dot
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf lisp
  neighbor 192.168.1.2 remote-as 65011
  neighbor 192.168.1.2 activate
 exit-address-family
 !
 address-family ipv6 vrf lisp
  neighbor 2001:db8:ffff::2 remote-as 65011
  neighbor 2001:db8:ffff::2 activate
 exit-address-family
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
!
ipv6 route ::/0 2001:db8:e000:2::f0f
!
```

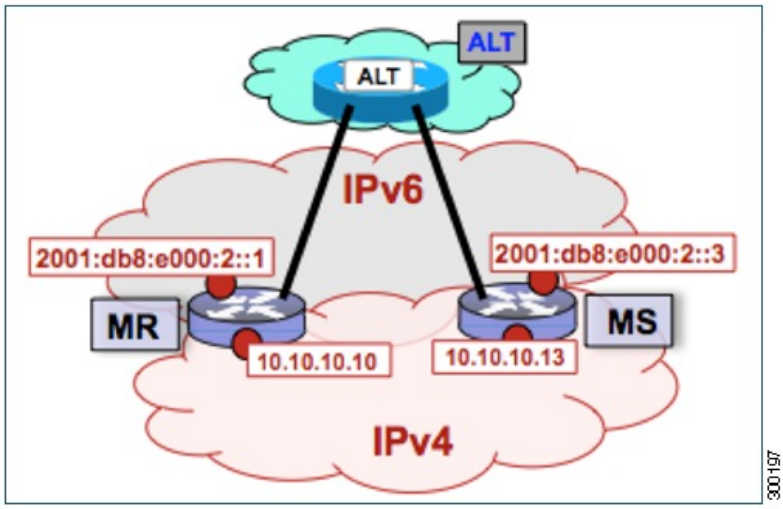
Configuring an ALT-Connected LISP Map Server

Perform this task to configure LISP alternative logical topology (ALT) map server functionality for both IPv4 and IPv6 address family mapping services.

Note You must also configure an ALT-connected LISP map resolver (see the Configuring an ALT-Connected LISP Map Resolver task).

In the figure below, the map resolver (MR) and map server (MS) are configured on separate devices and share their EID prefix information via connectivity.

Figure 161: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The map server illustrated in the topology shown in the figure is described below; the map resolver and LISP ALT are configured in separate tasks:

Mapping System

- Two LISP devices are configured, one as an MS and the other as an MR.
- The MS has an IPv4 locator of 10.10.10.13/24 and an IPv6 locator of 2001:db8:e000:2::3/64.
- The MR has an IPv4 locator of 10.10.10.10/24 and an IPv6 locator of 2001:db8:e000:2::1/64.
- Assume that the xTRs in the LISP site register to this map server. That is, the xTR registers the IPv4 EID-prefix of 172.16.1.0/24 and, when IPv6 EIDs are used, the xTR registers the IPv6 EID-prefix of 2001:db8:a::/48.

Note The configuration of the xTR must be changed to use the MS RLOC for its map server configuration and the MR RLOC for its map resolver configuration. For example:

- **ipv4 itr map-resolver 10.10.10.10**
- **ipv4 etr map-server 10.10.10.13 key 0 some-key**

Other Infrastructure

- The MR has IPv4 and IPv6 tunnel endpoints in the VRF table (named lisp) of 192.168.1.1/30 and 2001:db8:ffff::1/64, respectively, and the MS has IPv4 and IPv6 tunnel endpoints of 192.168.1.2/30 and 2001:db8:ffff::2/64, respectively, in the same VRF table. This tunnel is used for the ALT.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family ipv4** [**unicast**]
5. **exit-address-family**
6. **address-family ipv6**
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **ipv6 address** *ipv6-address/mask*
13. **tunnel source** *interface-type interface-number*
14. **tunnel destination** *ipv4-address*
15. **exit**
16. **router lisp**
17. **ipv4 map-server**
18. **ipv4 alt-vrf** *vrf-name*
19. **ipv6 map-server**
20. **ipv6 alt-vrf** *vrf-name*
21. **site** *site-name*
22. **eid-prefix** *EID-prefix*
23. **authentication-key** *key-type authentication-key*
24. **exit**
25. Repeat Steps 21 through 24 to configure additional LISP sites.
26. **exit**
27. **router bgp** *autonomous-system-number*
28. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
29. **redistribute lisp**
30. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
31. **neighbor** *ip-address* **activate**
32. **exit**
33. **address-family ipv6 vrf** *vrf-name*
34. **redistribute lisp**
35. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
36. **neighbor** *ip-address* **activate**
37. **exit**
38. **exit**
39. **ip route** *ipv4-prefix next-hop*

40. **ipv6 route** *ipv6-prefix next-hop*
41. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition lisp`	Creates a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF. In this example, a VRF named lisp is created to hold EID prefixes.
Step 3	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.
Step 4	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF table named lisp handles IPv4 EID prefixes.
Step 5	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv4 address family configuration mode and returns to VRF configuration mode.
Step 6	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv6 address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 9	**interface** *type number* **Example:** `Router(config)# interface tunnel 191`	Specifies the interface type of tunnel and the interface number and enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding lisp`	Associates a VRF instance configured in Step 2 with the tunnel interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.6 255.255.255.252`	Configures an IPv4 address for the tunnel interface.
Step 12	**ipv6 address** *ipv6-address/mask* **Example:** `Router(config-if)# ipv6 address 2001:DB8:ffff::6/64`	Configures an IPv6 address for the tunnel interface.
Step 13	**tunnel source** *interface-type interface-number* **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Configures the tunnel source.
Step 14	**tunnel destination** *ipv4-address* **Example:** `Router(config-if)# tunnel destination 10.10.10.13`	Configures the tunnel destination IPv4 address for the tunnel interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 16	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (Cisco IOS XE software only).
Step 17	**ipv4 map-server** **Example:**	Enables LISP map server functionality for EIDs in the IPv4 address family.

	Command or Action	Purpose
	`Router(config-router-lisp)# ipv4 map-server`	
Step 18	**ipv4 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv4 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv4 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 19	**ipv6 map-server** **Example:** `Router(config-router-lisp)# ipv6 map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family.
Step 20	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 21	**site** *site-name* **Example:** `Router(config-router-lisp)# site Site-1`	Specifies a LISP site and enters LISP site configuration mode. **Note** A LISP site name is locally significant to the map server on which it is configured. It has no relevance anywhere else. This name is used solely as an administrative means of associating one or more EID prefixes with an authentication key and other site-related mechanisms.
Step 22	**eid-prefix** *EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix 172.16.1.0/24`	Configures an IPv4 or IPv6 EID prefix associated with this LISP site. • Repeat this step as necessary to configure additional EID prefixes under this LISP sites. • In this step example, only an IPv4 EID prefix is configured but to complete the configuration, an IPv6 EID prefix must also be configured. **Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key. **Note** Additional **eid-prefix** command configuration options are available. (See the *LISP Command Reference* for more details.)
Step 23	**authentication-key** *key-type authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 some-key`	Configures the authentication key associated with this site. **Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key.

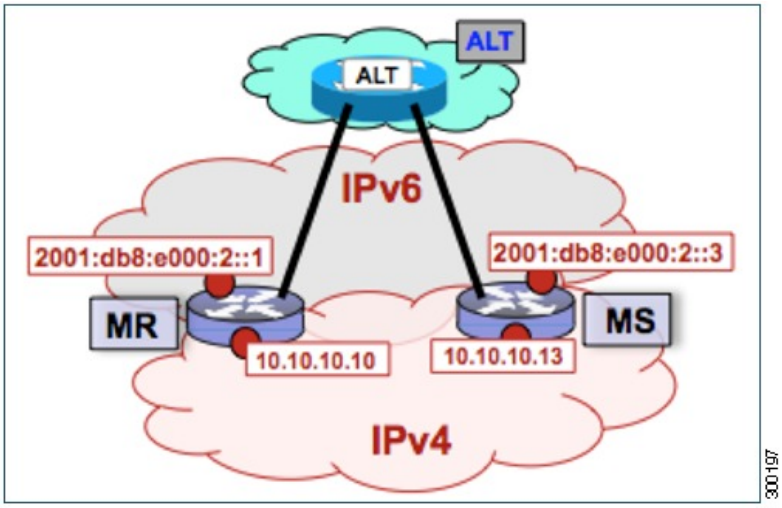
	Command or Action	Purpose
		Note The **authentication-key** can be configured with Type 6 encryption. (See the *LISP Command Reference* for more details.)
Step 24	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 25	Repeat Steps 21 through 24 to configure additional LISP sites.	—
Step 26	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 27	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65011`	Enters router configuration mode for the specified routing process.
Step 28	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf lisp`	Specifies the IPv4 address family and enters IPv4 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv4 VRF that carries EID prefixes in the LISP ALT.
Step 29	**redistribute lisp** **Example:** `Router(config-router-af)# redistribute lisp`	Redistributes EID prefixes known to LISP into BGP.
Step 30	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 65010`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 31	**neighbor** *ip-address* **activate** **Example:**	Enables the neighbor to exchange prefixes for the IPv4 unicast address family.

	Command or Action	Purpose
	`Router(config-router-af)# neighbor 192.168.1.1 activate`	
Step 32	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 33	**address-family ipv6 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv6 vrf lisp`	Specifies the IPv6 address family and enters IPv6 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv6 VRF that carries EID prefixes in the LISP ALT.
Step 34	**redistribute lisp** **Example:** `Router(config-router-af)# redistribute lisp`	Redistributes EID prefixes known to LISP into BGP.
Step 35	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::1 remote-as 65010`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 36	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::1 activate`	Enables the neighbor to exchange prefixes for the IPv6 unicast address family.
Step 37	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 38	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 39	**ip route** *ipv4-prefix next-hop* **Example:**	Configures an IPv4 static route.

	Command or Action	Purpose
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	• In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 40	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 41	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Example:

Figure 162: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a LISP map server including some basic IP and IPv6 configuration not included in the task table for this task:

```
!
hostname MS
!
vrf definition lisp
 rd 1:1
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no ip domain lookup
```

```
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface Tunnel192
 vrf forwarding lisp
 ip address 192.168.1.2 255.255.255.252
 ipv6 address 2001:db8:ffff::2/64
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.10.10.10
!
interface GigabitEthernet 0/0/0
 description Link to SP1 (RLOC)
 ip address 10.10.10.13 255.255.255.0
 ipv6 address 2001:db8:e000:2::3/64
!
router lisp
 site Site-1
  authentication-key 0 some-xtr-key
  eid-prefix 172.16.1.0/24
  eid-prefix 2001:db8:a::/48
  exit
 !
 site Site-2
  authentication-key 0 another-xtr-key
  eid-prefix 172.16.2.0/24
  eid-prefix 2001:db8:b::/48
  exit
 !
 !---configure more LISP sites as required---
 !
 ipv4 map-server
 ipv4 alt-vrf lisp
 ipv6 map-server
 ipv6 alt-vrf lisp
 exit
!
router bgp 65011
 bgp asnotation dot
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf lisp
  redistribute lisp
  neighbor 192.168.1.1 remote-as 65010
  neighbor 192.168.1.1 activate
 exit-address-family
 !
 address-family ipv6 vrf lisp
  redistribute lisp
  neighbor 2001:db8:ffff::1 remote-as 65010
  neighbor 2001:db8:ffff::1 activate
 exit-address-family
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
!
ipv6 route ::/0 2001:db8:e000:2::f0f
```

Configure a PETR and a PITR

The following tasks show how to design and deploy a Proxy Egress Tunnel Router (PETR) and a Proxy Ingress Tunnel Router (PITR). The example scenario shows deployment of a PETR and PITR as separate devices but it is also possible to deploy a single device that acts simultaneously as a PETR and a PITR, which is called a PxTR.

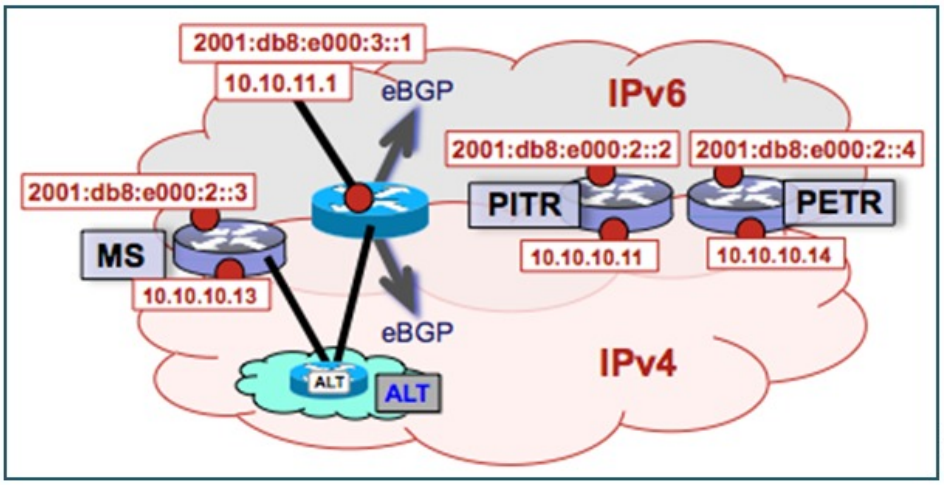
Deploying a Proxy Egress Tunnel Router with both an IPv4 and an IPv6 RLOC

Perform this task to deploy a Proxy Egress Tunnel Router (PETR) for both IPv4 and IPv6 address families. You can also perform this task to configure PETR functionality on a single device that acts simultaneously as a PETR and as a Proxy Ingress Tunnel Router (PITR), referred to as a PxTR.

A PETR simply takes in LISP encapsulated packets and decapsulates them and forwards them. For example, a PETR can be used to provide IPv6 LISP EIDs access to non-LISP EIDs when the LISP site only has IPv4 RLOC connectivity. A PETR, therefore, is used for LISP-to-non-LISP access in situations where cross-address family connectivity is an issue. (A PETR can still be used for matching EID and RLOC address families if desired.) Note that a PITR is required to provide return-traffic flow. A PETR is simple to deploy because it need only provide dual-stack connectivity to the core.

The topology used in this PETR example is shown in the figure. The PETR and PITR in this example are deployed as separate devices and each have both an IPv4 and an IPv6 locator.

Figure 163: Proxy Egress Tunnel Router with both an IPv4 and an IPv6 RLOC



The components illustrated in the topology shown in the figure are described below:

PETR

- When deployed as a standalone LISP device, the PETR has dual-stack connectivity to the core network.
- The PETR IPv4 locator is 10.10.10.14/24 and the IPv6 locator is 2001:db8:e000:2::4/64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**

4. **ipv4 proxy-etr**
5. **ipv6 proxy-etr**
6. **exit**
7. **ip route** *ipv4-prefix next-hop*
8. **ipv6 route** *ipv6-prefix next-hop*
9. **exit**

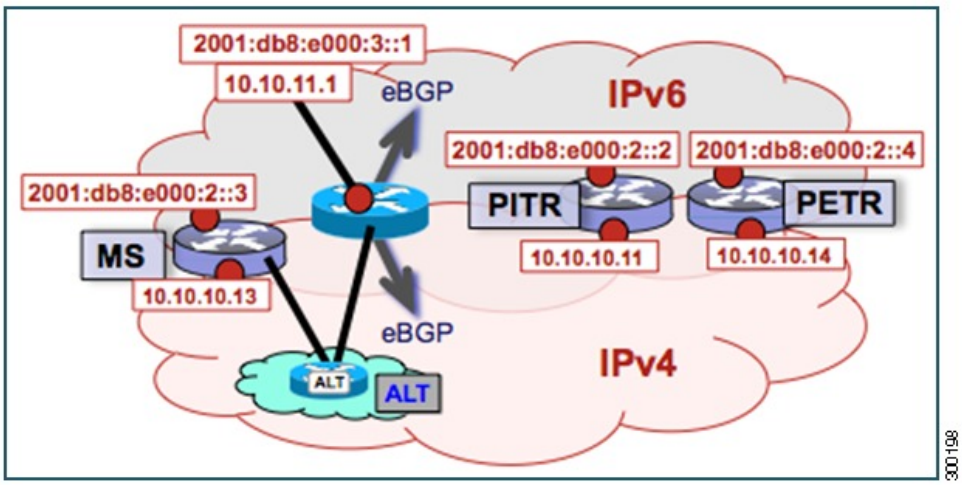
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 4	**ipv4 proxy-etr** **Example:** `Router(config-router-lisp)# ipv4 proxy-etr`	Enables PETR functionality for IPv4 EIDs.
Step 5	**ipv6 proxy-etr** **Example:** `Router(config-router-lisp)# ipv6 proxy-etr`	Enables PETR functionality for IPv6 EIDs.
Step 6	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and enters global configuration mode.
Step 7	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 8	**ipv6 route** *ipv6-prefix next-hop* **Example:**	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.

	Command or Action	Purpose
	`Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Example:

Figure 164: Proxy Egress Tunnel Router with both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a PETR including some basic IP and IPv6 configuration not included in the task table for this task:

```
!
hostname PETR
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface GigabitEthernet 0/0/0
 description Link to Core (RLOC)
 ip address 10.10.10.14 255.255.255.0
 ipv6 address 2001:db8:e000:2::4/64
!
router lisp
 ipv4 proxy-etr
 ipv6 proxy-etr
 exit
!
```

```
ip route 0.0.0.0 0.0.0.0 10.10.10.1
!
ipv6 route ::/0 2001:db8:e000:2::f0f
```

Deploying a Proxy Ingress Tunnel Router with both an IPv4 and an IPv6 RLOC

Perform this task to deploy a Proxy Ingress Tunnel Router (PITR) for both IPv4 and IPv6 address families. You can also perform this task to configure PITR functionality on a single device that acts simultaneously as a PITR and as a Proxy Egress Tunnel Router (PETR), referred to as a PxTR.

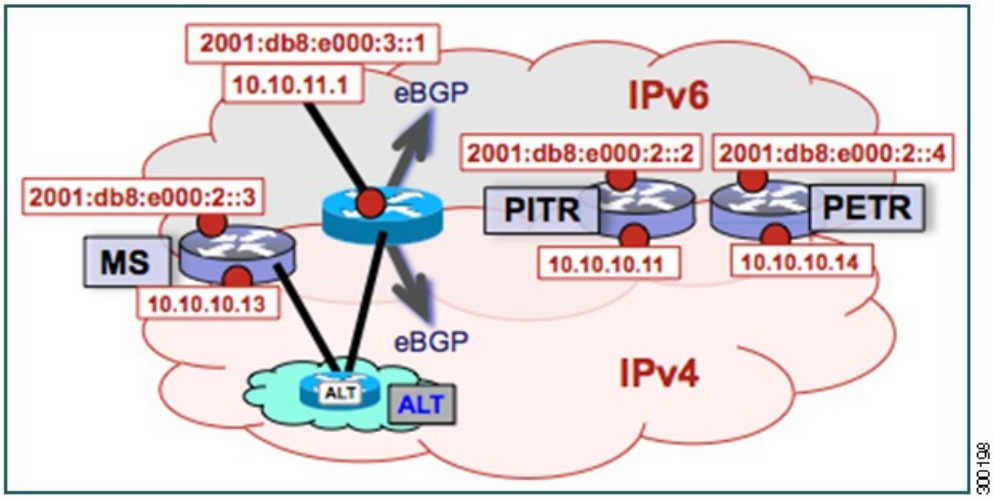
A PITR attracts non-LISP packets by advertising a coarse-aggregate prefix for LISP EIDs into the core (such as the Internet or a Multiprotocol Label Switching (MPLS) core) and then performs LISP encapsulation services (like an ITR) to provide access to LISP EIDs. Thus, a PITR provides non-LISP-to-LISP interworking. A PITR is also used to provide address family “hop-over﷯? for non-LISP-to-LISP traffic. For example, a dual-stacked PxTR can be used to provide a return-traffic path from non-LISP IPv6 sites to IPv6 LISP sites that contain only IPv4 RLOCs.

To resolve EID-to-RLOC mappings for creating non-LISP-to-LISP flows, configure PITR to query the LISP mapping system. In this task, the PITR is configured to send Map-Rrequest messages via the LISP alternate logical topology (ALT) to resolve EID-to-RLOC mappings.

Note To attract non-LISP traffic destined to LISP sites, the PITR must advertise coarse-aggregate EID prefixes into the underlying network infrastructure. In an Internet-as-the-core example, attracting non-LISP traffice destined to LISP sites is typically managed via external BGP (eBGP) and by advertising the coarse-aggregate that includes all appropriate EID prefixes into the Internet. The example configuration in the figure utilizes this approach. Because this is a standard BGP configuration, summary and detailed command guidance is not provided in the task table for this task, although the complete configuration example that follows the task table does include an accurate example of this eBGP peering. Any other approach that advertises coarse-aggregates that include all appropriate EID prefixes into the core are also acceptable.

The topology used in this example is shown in the figure. The PITR is deployed as a separate device, with both an IPv4 and an IPv6 locator. A map resolver and core-peering router are also shown in the figure for reference because they are required components for completing the PITR configuration shown in the figure.

Figure 165: Proxy Ingress Tunnel Router with both an IPv4 and an IPv6 RLOC



The components illustrated in the topology shown in the figure are described below:

PITR

- When deployed as a standalone LISP device, the PITR has dual-stack connectivity to the core network.
- The PITR IPv4 locator is 10.10.10.11/24 and the IPv6 locator is 2001:db8:e000:2::2/64.
- The use of LISP EID prefixes throughout this task (172.16.1.0/24 and 2001:db8:a::/48 configuration) is assumed and are part of LISP EID blocks that can be summarized in coarse-aggregates and advertised by the PITR into the core network. The advertisement of the IPv4 coarse-aggregate of 172.16.0.0/16 and the IPv6 coarse-aggregate of 2001:db8::/33 by the PITR into the IPv4 and IPv6 core networks is also assumed.
- The PITR eBGP peers with the core router with locators 10.10.11.1 and 2001:db8:e000:3::1 in order to advertise the coarse-aggregate IPv4 EID prefix of 172.16.0.0/16 and the IPv6 EID prefix of 2001:db8::/33 into the IPv4 and IPv6 cores, respectively.
- The PITR is configured to use the LISP ALT (GRE+BGP) via the map server with locators 10.10.10.13 and 2001:db8:e000:2::3. The relevant configuration is shown for the PITR.

Other Infrastructure

- The MS has IPv4 and IPv6 tunnel endpoints in the VRF table (named lisp) of 192.168.5/30 and 2001:db8:ffff::5/64, respectively. The configuration of the map server is not in the task table.
- The core router has an IPv4 address of 10.10.11.1 and an IPv6 address of 2001:db8:e000:3::1. These addresses will be used for eBGP peering. The core router configuration is assumed to be familiar as a typical ISP peering router and is therefore not included in the task table.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family ipv4** [**unicast**]

5. **exit-address-family**
6. **address-family ipv6**
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **ipv6 address** *ipv6-address/mask*
13. **tunnel source** *interface-type interface-number*
14. **tunnel destination** *ipv4-address*
15. **exit**
16. **router lisp**
17. **ipv4 alt-vrf** *vrf-name*
18. **ipv4 proxy-itr** *ipv4-locator* [*ipv6-locator*]
19. **ipv4 map-cache-limit** *map-cache-limit*
20. **ipv6 alt-vrf** *vrf-name*
21. **ipv6 proxy-itr** *ipv6-locator* [*ipv4-locator*]
22. **ipv6 map-cache-limit** *map-cache-limit*
23. **exit**
24. **router bgp** *autonomous-system-number*
25. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
26. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
27. **neighbor** *ip-address* **activate**
28. **exit**
29. **address-family ipv6** [**unicast** | **multicast** | **vrf** *vrf-name*]
30. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
31. **neighbor** *ip-address* **activate**
32. **exit**
33. **exit**
34. **ip route** *ipv4-prefix next-hop*
35. **ip route** *ipv4-prefix next-hop*
36. **ipv6 route** *ipv6-prefix next-hop*
37. **ipv6 route** *ipv6-prefix next-hop*
38. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:**	Configures a virtual routing and forwarding (VRF) table and enters VRF configuration mode.

	Command or Action	Purpose
	`Router(config)# vrf definition lisp`	• Use the *vrf-name* argument to specify a name to be assigned to the VRF. In this example, a VRF named lisp is created to hold EID prefixes.
Step 3	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.
Step 4	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF named lisp handles IPv4 EID prefixes.
Step 5	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 6	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface tunnel 191`	Specifies the interface type of tunnel and the interface number and enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding lisp`	Associates a VRF instance configured in Step 2 with the tunnel interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.

	Command or Action	Purpose
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.6 255.255.255.252`	Configures an IPv4 address for the tunnel interface.
Step 12	**ipv6 address** *ipv6-address/mask* **Example:** `Router(config-if)# ipv6 address 2001:DB8:ffff::6/64`	Configures an IPv6 address for the tunnel interface.
Step 13	**tunnel source** *interface-type interface-number* **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Configures the tunnel source.
Step 14	**tunnel destination** *ipv4-address* **Example:** `Router(config-if)# tunnel destination 10.10.10.13`	Configures the tunnel destination IPv4 address for the tunnel interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 16	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 17	**ipv4 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv4 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv4 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 18	**ipv4 proxy-itr** *ipv4-locator* [*ipv6-locator*] **Example:** `Router(config-router-lisp)# ipv4 proxy-itr 10.10.10.11 2001:db8:e000:2::2`	Enables Proxy Ingress Tunnel Router (PITR) functionality for IPv4 EIDs, and specifies the IPv4 and (optionally) the IPv6 RLOCs (local to the PITR) to use when LISP-encapsulating packets to LISP sites.
Step 19	**ipv4 map-cache-limit** *map-cache-limit* **Example:**	Specifies the maximum number of IPv4 map-cache entries to be maintained by the PITR. • When the map-cache reaches this limit, existing entries are removed according to the rules described

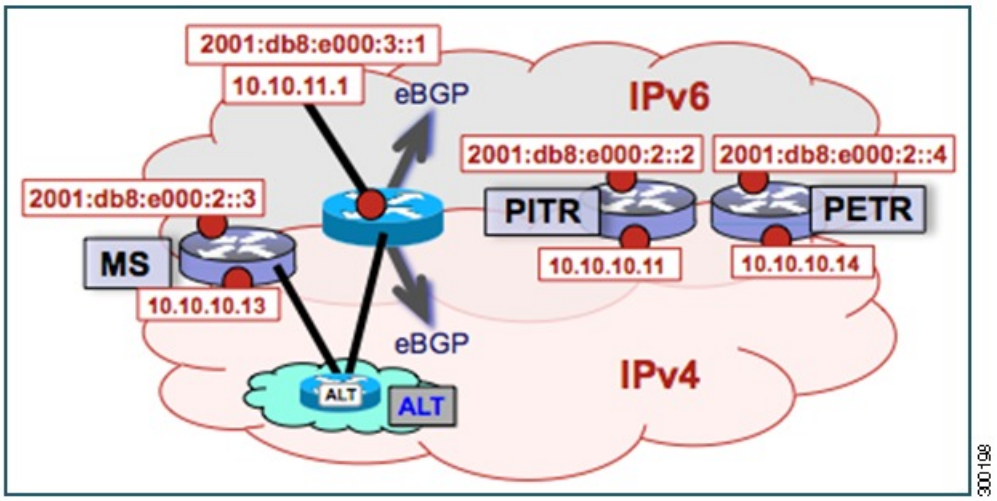
	Command or Action	Purpose
	`Router(config-router-lisp)# ipv4 map-cache-limit 100000`	in the command reference guide. (See the *LISP Command Reference* for more details.) • The default map-cache-limit is 10000. In this example, since the device is being configured as a PITR, a larger map-cache limit is configured.
Step 20	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 21	**ipv6 proxy-itr** *ipv6-locator* [*ipv4-locator*] **Example:** `Router(config-router-lisp)# ipv6 proxy-itr 2001:db8:e000:2::2 10.10.10.11`	Enables Proxy Ingress Tunnel Router (PITR) functionality for IPv6 EIDs, and specifies the IPv6 and (optionally) the IPv4 RLOCs (local to the PITR) to use when LISP-encapsulating packets to LISP sites.
Step 22	**ipv6 map-cache-limit** *map-cache-limit* **Example:** `Router(config-router-lisp)# ipv6 map-cache-limit 100000`	Specifies the maximum number of IPv6 map-cache entries to be maintained by the PITR. • When the map-cache reaches this limit, existing entries are removed according to the rules described in the command reference guide. (See the *LISP Command Reference* for more details.) The default map-cache-limit is 10000. In this example, since the device is being configured as a PITR, a larger map-cache limit is configured.
Step 23	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 24	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65015`	Enters router configuration mode for the specified routing process.
Step 25	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf lisp`	Specifies the IPv4 address family and enters IPv4 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv4 VRF that carries EID prefixes in the LISP ALT.

	Command or Action	Purpose
Step 26	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.5 remote-as 65011`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 27	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.5 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family.
Step 28	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode.
Step 29	**address-family ipv6** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router-af)# address-family ipv6 vrf lisp`	Specifies the IPv6 address family and enters IPv6 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv6 VRF that carries EID prefixes in the LISP ALT.
Step 30	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::5 remote-as 65011`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 31	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::5 activate`	Enables the neighbor to exchange prefixes for the IPv6 unicast address family.
Step 32	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode.
Step 33	**exit** **Example:**	Exits router configuration mode.

	Command or Action	Purpose
	`Router(config-router)# exit`	
Step 34	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 35	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 172.16.0.0 255.255.0.0 Null0 tag 123`	Configures an IPv4 static route. • In this example, a static route is configured to Null0 for the coarse-aggregate IPv4 EID prefix 172.16.0.0/16. This static route is required to ensure proper operation of LISP in querying the mapping system for LISP EIDs. The tag 123 is added to this null route as a reference point for the route map used to permit the advertisement of this coarse aggregate to the upstream ISP BGP peer.
Step 36	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 37	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route 2001:db8::/33 Null0 tag 123`	Configures an IPv6 static route. • In this example, a static route is configured to Null0 for the coarse-aggregate IPv6 EID prefix 2001:db8::/33. This is required to ensure proper operation of LISP in querying the mapping system for LISP EIDs. The tag 123 is added to this null route as a handy reference point for the route-map used to permit the advertisement of this coarse-aggregate to the upstream ISP BGP peer.
Step 38	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 166: Proxy Ingress Tunnel Router with both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a PITR includes some basic IP, BGP, and route map configuration not included in the task table for this task:

```
!
hostname PITR
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface Tunnel191
 vrf forwarding lisp
 ip address 192.168.1.6 255.255.255.252
 ipv6 address 2001:db8:ffff::6/64
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.10.10.13
!
interface GigabitEthernet 0/0/0
 description Link to Core (RLOC)
 ip address 10.10.10.11 255.255.255.0
 ipv6 address 2001:db8:e000:2::2/64
!
router lisp
 ipv4 alt-vrf lisp
 ipv4 map-cache-limit 100000
 ipv4 proxy-itr 10.10.10.11 2001:db8:e000:2::2
 ipv6 alt-vrf lisp
 ipv6 map-cache-limit 100000
 ipv6 proxy-itr 2001:db8:e000:2::2 10.10.10.11
 exit
!
router bgp 65015
 bgp asnotation dot
 bgp log-neighbor-changes
```

```
 neighbor 10.10.11.1 remote-as 65111
 neighbor 2001:db8:e000:3::1 remote-as 65111
 !
 address-family ipv4
  no synchronization
  redistribute static route-map populate-default
  neighbor 10.10.11.1 activate
  neighbor 10.10.11.1 send-community both
  neighbor 10.10.11.1 route-map dfz-out out
 exit-address-family
 !
 address-family ipv6
  redistribute static route-map populate-default
  neighbor 2001:db8:e000:3::1  activate
  neighbor 2001:db8:e000:3::1  send-community both
  neighbor 2001:db8:e000:3::1  route-map dfz-out out
 exit-address-family
 !
 address-family ipv4 vrf lisp
  no synchronization
  neighbor 192.168.1.5 remote-as 65011
  neighbor 192.168.1.5 activate
 exit-address-family
 !
 address-family ipv6 vrf lisp
  no synchronization
  neighbor 2001:db8:ffff::5 remote-as 65011
  neighbor 2001:db8:ffff::5 activate
 exit-address-family
!
ip bgp-community new-format
ip community-list standard dfz-upstream permit 65100:123
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
ip route 172.16.0.0 255.255.0.0 Null0 tag 123
!
ipv6 route 2001:db8::/33 Null0 tag 123
ipv6 route ::/0 2001:db8:e000:2::f0f
!
route-map populate-default permit 10
 match tag 123
 set origin igp
 set community 65100:123
!
route-map dfz-out permit 10
 match community dfz-upstream
!
```

Verify and Troubleshoot Locator ID Separation Protocol

Once LISP is configured, you can verify and troubleshoot LISP configuration and operations by following the optional steps in this task. Note that certain verification and troubleshooting steps are specific to certain LISP devices and only apply if configured in your LISP site. For the below commands, if **instance-id** is unspecified, the ID is specified as zero.

SUMMARY STEPS

1. **enable**
2. **show running-config | section router lisp**

3. **show** [**ip** | **ipv6**] **lisp** [**instance-id** *number*]
4. **show** [**ip** | **ipv6**] **lisp map-cache** [**instance-id** *number*]
5. **show** [**ip** | **ipv6**] **lisp database** [**instance-id** *number*]
6. **show lisp site** [**name** *site-name*]
7. **lig** {[**instance-id** *number*] [**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}}
8. **ping** {*hostname* | *destination-EID*}
9. **clear** [**ip** | **ipv6**] **lisp map-cache** [**instance-id** *number* | *****]

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 show running-config | section router lisp

The **show running-config | section router lisp** command is useful for quickly verifying the LISP configuration on the device. This command applies to any Cisco IOS LISP device.

The following is sample output from the **show running-config | section router lisp** command when a mulithomed LISP site is configured with IPv4 and IPv6 EID prefixes:

Example:

```
Router# show running-config | section router lisp
router lisp
 service ipv4
  itr map-resolver 10.10.10.10
  itr map-resolver 10.10.30.10
  itr
  etr map-server 10.10.10.10 key some-key
  etr map-server 10.10.30.10 key some-key
  etr
  exit-service-ipv4
 !
 service ipv6
  itr map-resolver 10.10.10.10
  itr map-resolver 10.10.30.10
  itr
  etr map-server 10.10.10.10 key some-key
  etr map-server 10.10.30.10 key some-key
  etr
  use-petr 10.10.10.11
  use-petr 10.10.30.11
  exit-service-ipv6
 !
 instance-id 1
  service ipv4
   eid-table default
   database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
   database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
   exit-service-ipv4
  !
  service ipv6
   eid-table default
   database-mapping 2001:DB8:A::/48 10.1.1.2 priority 1 weight 50
```

```
   database-mapping 2001:DB8:A::/48 10.2.1.2 priority 1 weight 50
   exit-service-ipv6
  !
  exit-instance-id
 !
 exit-router-lisp
```

Step 3 **show** [**ip** | **ipv6**] **lisp** [**instance-id** *number*]

The **show ip lisp** and **show ipv6 lisp** commands are useful for quickly verifying the operational status of LISP as configured on the device, as applicable to the IPv4 and IPv6 address families, respectively. This command applies to any Cisco IOS LISP device.

Example:

The following example shows LISP operational status and IPv4 address family information:

```
Router# show ip lisp

Ingress Tunnel Router (ITR):      enabled
  Egress Tunnel Router (ETR):       enabled
  Proxy-ITR Router (PITR):          disabled
  Proxy-ETR Router (PETR):          disabled
  Map Server (MS):                  disabled
  Map Resolver (MR):                disabled
  Map-Request source:               172.16.1.1
  ITR Map-Resolver(s):              10.10.10.10, 10.10.30.10
  ETR Map-Server(s):                10.10.10.10 (00:00:56), 10.10.30.10 (00:00:12)
  ETR accept mapping data:          disabled, verify disabled
  ETR map-cache TTL:                1d00h
  Locator Status Algorithms:
    RLOC-probe algorithm:           disabled
  Static mappings configured:       0
  Map-cache size/limit:             2/1000
  Map-cache activity check period:  60 secs
  Map-database size:                1
```

Example:

The following example shows LISP operational status and IPv6 address family information:

```
Router# show ip lisp

Ingress Tunnel Router (ITR):      enabled
  Egress Tunnel Router (ETR):       enabled
  Proxy-ITR Router (PITR):          disabled
  Proxy-ETR Router (PETR):          disabled
  Map Server (MS):                  disabled
  Map Resolver (MR):                disabled
  Map-Request source:               2001:DB8:A::1
ITR Map-Resolver(s):                10.10.10.10, 10.10.30.10
  ETR Map-Server(s):                10.10.10.10 (00:00:23), 10.10.30.10 (00:00:40)
  ETR accept mapping data:          disabled, verify disabled
  ETR map-cache TTL:                1d00h
  Locator Status Algorithms:
    RLOC-probe algorithm:           disabled
  Static mappings configured:       0
  Map-cache size/limit:             1/1000
  Map-cache activity check period:  60 secs
  Map-database size:                1
```

Step 4 **show** [**ip** | **ipv6**] **lisp map-cache** [**instance-id** *number*]

The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands are useful for quickly verifying the operational status of the map-cache on a device configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families, respectively. Based on a configuration when a mulithomed LISP site is configured with IPv4 and IPv6 EID prefixes, this example output assumes that a map-cache entry has been received for another site with the IPv4 EID prefix of 172.16.2.0/24 and the IPv6 EID prefix of 2001:db8:b::/48.

Example:

The following example shows IPv4 mapping cache information:

```
Router# show ip lisp map-cache

LISP IPv4 Mapping Cache, 2 entries

0.0.0.0/0, uptime: 02:48:19, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
172.16.2.0/24, uptime: 01:45:24, expires: 22:14:28, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  01:45:24  up           1/1
```

Example:

The following example shows IPv6 mapping cache information:

```
Router# show ipv6 lisp map-cache

LISP IPv6 Mapping Cache, 2 entries

::/0, uptime: 02:49:39, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2001:DB8:B::/48, uptime: 00:00:07, expires: 23:59:46, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  00:00:07  up           1/1
```

Step 5 **show** [**ip** | **ipv6**] **lisp database** [**instance-id** *number*]

The **show ip lisp database** and **show ipv6 lisp database** commands are useful for quickly verifying the the operational status of the database mapping on a device configured as an ETR, as applicable to the IPv4 and IPv6 address families, respectively. The following example output is based on a configuration when a mulithomed LISP site is configured with IPv4 and IPv6 EID prefixes.

Example:

The following example shows IPv4 mapping database information:

```
Router# show ip lisp database

LISP ETR IPv4 Mapping Database, LSBs: 0x3, 1 entries

172.16.1.0/24
```

Example:

The following example shows IPv6 mapping database information:

```
Router# show ipv6 lisp database

LISP ETR IPv6 Mapping Database, LSBs: 0x1, 1 entries

2001:DB8:A::/48
```

Step 6 **show lisp site** [**name** *site-name*]

The **show lisp site** command is useful for quickly verifying the operational status of LISP sites, as configured on a map server. This command applies only to a device configured as a map server.

The following examples are based on configurations where a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes:

Example:

```
Router# show lisp site

LISP Site Registration Information

Site Name      Last      Up   Who Last            EID Prefix
               Register       Registered
Site-1         00:00:15  yes  10.1.1.2            172.16.1.0/24
               00:00:11  yes  10.1.1.2            2001:DB8:A::/48
Site-2         00:00:27  yes  10.0.0.6            172.16.2.0/24
               00:00:37  yes  10.0.0.6            2001:DB8:B::/48
```

Example:

```
Router# show lisp site name Site-1

Site name: Site-1
Allowed configured locators: any
Allowed EID-prefixes:
  EID-prefix: 172.16.1.0/24
    First registered:     00:04:51
    Routing table tag:    0
    Origin:               Configuration
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 10.1.1.2, last registered 00:00:01, no proxy-reply, map-notify
                 TTL 1d00h, no merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2  yes   up          1/50
    ETR 10.2.1.2, last registered 00:00:03, no proxy-reply, map-notify
                 TTL 1d00h, merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2 yes    up          1/50
      10.2.1.2 yes    up          1/50
  EID-prefix: 2001:DB8:A::/48
    First registered:     00:04:51
    Routing table tag:    0
    Origin:               Configuration
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 10.1.1.2, last registered 00:00:01, no proxy-reply, map-notify
                 TTL 1d00h, no merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2  yes   up          1/50
    ETR 10.2.1.2, last registered 00:00:03, no proxy-reply, map-notify
                 TTL 1d00h, merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2 yes    up          1/50
      10.2.1.2 yes    up          1/50
```

Step 7 **lig** {[**instance-id** *number*] [**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}}

The LISP Internet Groper (lig) command is useful for testing the LISP control plane. The **lig** command can be used to query for the indicated destination hostname or EID, or the router's local EID prefix. This command provides a simple means of testing whether a destination EID exists in the LISP mapping database system, or whether your site is registered with the mapping database system. This command is applicable for both the IPv4 and IPv6 address families and applies to any Cisco IOS LISP device that maintains a map-cache (i.e. configured as an ITR or PITR).

The following examples are based on configurations where a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes:

Example:

```
Router# lig self ipv4

Mapping information for EID 172.16.1.0 from 10.1.1.2 with RTT 12 msecs
172.16.1.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.1.1.2  00:00:00  up, self     1/50
  10.2.1.2  00:00:00  up           1/50
```

Example:

```
Router# lig self ipv6

Mapping information for EID 2001:DB8:A:: from 10.0.0.2 with RTT 12 msecs
2001:DB8:A::/48, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.1.1.2  00:00:00  up, self     1/50
  10.2.1.2  00:00:00  up           1/50
```

Example:

```
Router# lig 172.16.2.1

Mapping information for EID 2001:DB8:A:: from 10.0.0.2 with RTT 12 msecs
2001:DB8:A::/48, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.1.1.2  00:00:00  up, self     1/50
  10.2.1.2  00:00:00  up           1/50
```

Example:

```
Router# lig 2001:db8:b::1

Mapping information for EID 172.16.2.1 from 10.0.0.6 with RTT 4 msecs
2001:DB8:B::/48, uptime: 01:52:45, expires: 23:59:52, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  01:52:45  up           1/1
```

Step 8 **ping** {*hostname* | *destination-EID*}

The **ping** command is useful for testing basic network connectivity and reachability and liveness of a destination EID or RLOC address. It is important to be aware that because LISP uses encapsulation, you should always specify a source address when using **ping**. Never allow the **ping** application to assign its own default source address because there are four possible ways to use **ping** and unless the source address is explicitly named, the wrong address may be used by the application and return erroneous results that complicate operational verification or troubleshooting.

The four possible uses of **ping** are:

- RLOC-to-RLOC—Sends out “echo?? packets natively (no LISP encapsulation) and receives the “echo-reply?? back natively. This use of **ping** can test the underlying network connectivity between locators of various devices, such as between an xTR and a map server or map resolver.

- EID-to-EID—Sends out “echo?? packets with LISP encapsulation and receives the “echo-reply?? back as LISP encapsulated. This use of **ping** can be used to test the LISP data plane (encapsulation) between LISP sites.
- EID-to-RLOC—Sends out “echo?? packets natively (no LISP encapsulation) and receives the "echo-reply" back as LISP encapsulated through a PITR mechanism. This use of **ping** can be used to test the PITR infrastructure.
- RLOC-to-EID - Sends out “echo?? packets with LISP encapsulation and receives the “echo-reply?? back natively (no LISP encapsulation. This use of **ping** can be used to test PETR capabilities.

The **ping** command is applicable to the IPv4 and IPv6 address families, respectively, and can be used on any IOS XE LISP device but is limited by the LISP device and site configuration. (For example, the ability to do LISP encapsulation requires the device to be configured as either an ITR or PITR.)

The following examples are based on configurations where a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes:

Example:

```
Router# ping 172.16.2.1 source 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms
```

Example:

```
Router# ping 2001:db8:b::1 source 2001:db8:a::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:A::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms
```

Step 9 **clear** [**ip** | **ipv6**] **lisp map-cache** [**instance-id** *number* | *****]

The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries stored by the router. The ***** keyword clears all entries. This command applies to a LISP device that maintains a map-cache (like one configured as an ITR or PITR) and can be useful if trying to quickly verify the operational status of the LISP control plane. Based on a configuration when a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes, the following example output assumes that a map-cache entry has been received for another site with the IPv4 EID prefix of 172.16.2.0/24 or an IPv6 EID prefix of 2001:db8:b::/48.

Example:

The following example shows IPv4 mapping cache information, how to clear the mapping cache, and the **show** information after the cache is cleared.

```
Router# show ip lisp map-cache

LISP IPv4 Mapping Cache, 2 entries

0.0.0.0/0, uptime: 02:48:19, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
172.16.2.0/24, uptime: 01:45:24, expires: 22:14:28, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  01:45:24  up           1/1

Router# clear ip lisp map-cache

Router# show ip lisp map-cache
```

```
LISP IPv4 Mapping Cache, 1 entries

0.0.0.0/0, uptime: 00:00:02, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

Example:

The following example shows IPv6 mapping cache information, how to clear the mapping cache, and the **show** information after the cache is cleared.

```
Router# show ipv6 lisp map-cache

LISP IPv6 Mapping Cache, 2 entries

::/0, uptime: 02:49:39, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2001:DB8:B::/48, uptime: 00:00:07, expires: 23:59:46, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  00:00:07  up           1/1

Router# clear ip lisp map-cache

Router# show ip lisp map-cache

LISP IPv6 Mapping Cache, 1 entries

::/0, uptime: 00:00:02, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

Additional References for Configuring LISP

The following sections provide references related to the Locator ID Separation Protocol.

Related Documents

Document Title	Location
Cisco IOS LISP Lab Test Configuration Application Note	http://lisp4.cisco.com/lisp_tech.html
Cisco IOS IP Routing: LISP Command Reference	http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_lisp/command/ip-lisp-cr-book.html

MIBs

MIB	MIBs Link
LISP MIB	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 6830	Locator/ID Separation Protocol (LISP) http://tools.ietf.org/html/draft-ietf-lisp-07
RFC 6832	Interworking LISP with IPv4 and IPv6 https://tools.ietf.org/html/rfc6832
RFC 6833	LISP Map Server https://tools.ietf.org/html/rfc6833
RFC 6835	LISP Internet Groper (LIG) https://tools.ietf.org/html/rfc6835

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 192: Feature Information for Locator/ID Separation Protocol

Feature Name	Release	Feature Configuration Information
Configure LISP		Introduces LISP functionality to support ITR, ETR, PITR, PETR, MS, MR, and LISP ALT devices for IPv4 and IPv6 address families.

Feature Name	Release	Feature Configuration Information
LISP MIB		This feature introduces LISP MIB on Cisco software.

CHAPTER 179

LISP Multicast

The LISP Multicast feature introduces support for carrying multicast traffic over a Locator ID Separation Protocol (LISP) overlay. This support currently allows for unicast transport of multicast traffic with head-end replication at the root ingress tunnel router (ITR) site. This allows network operators to use LISP to carry multicast traffic over core networks that do not have native multicast capabilities.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for LISP Multicast

- You must configure basic LISP services on the device. Basic LISP configurations are covered in "Configuring Basic LISP" section of this configuration guide.
- You must configure IPv6 multicast and LISP services on the device. The configuration of IPv6 multicast over LISP is covered in "How to Configure LISP Multicast" and "Example: Configuring IPv6 Multicast over LISP" sections of this guide.

Restrictions for LISP Multicast

- LISP multicast does not support IPv6 endpoint identifiers (EIDs) or IPv6 routing locators (RLOCs). Only IPv4 EIDs and IPv4 RLOCs are supported.
- LISP multicast does not support Dense Mode or Bidirectional Protocol Independent Multicast (PIM). Only PIM-Sparse Mode (SM) and PIM Source Specific Multicast (SSM) modes are supported.
- LISP multicast does not support group to Rendezvous Point (RP) mapping distribution mechanisms, Auto-RP and Bootstrap Router (BSR). Only static-RP configuration is supported.
- LISP multicast does not support LISP Virtual Machine Mobility (VM-Mobility) deployment. That is, LISP multicast cannot be used as a data center interconnect (DCI) mechanism.
- IPv6 LISP multicast does not support IPv6 routing locators. Additionally, it does not support multicast transport.

Note IPv6 LISP multicast is supported only from Cisco IOS Release 16.2 onwards, though releases earlier than 16.2 supports only IPv4 LISP multicast

Information About LISP Multicast

The implementation of LISP multicast includes the following features:

- Mapping of multicast source addresses as LISP endpoint identifiers (EIDs). (Destination group addresses are not topology dependent).
- Building the multicast distribution tree across LISP overlays.
- Unicast head-end replication of multicast data packets from sources within a root ingress tunnel router (ITR) site to receiver egress tunnel routers (ETRs).
- Support for ASM (Any Source Multicast) and SSM (Source Specific Multicast).
- Support for various combinations of LISP and non-LISP capable source and receiver sites.
- Support for IPv6 endpoint identifiers (EIDs).

Note If a LISP xTR is also a PIM First Hop Router (FH) or a Rendezvous Point (RP) and the device is only receiving traffic, ensure that at least one interface on the device is covered by a local LISP database mapping. No additional configuration is required to ensure that proper address is selected.

How to Configure LISP Multicast

Configuring LISP Multicast

Perform this task to enable the LISP multicast functionality on the xTR.

Before you begin

Ensure that generic multicast functionality has been enabled on the required devices of the LISP site and PIM sparse mode has been enabled on the required interfaces of these devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**distributed**]
4. Enter one of the following:
   - **ip pim rp-address** *rp-address*
   - **ip pim ssm** {**default** | **range** {*access-list-number* | *access-list-name*}}
5. **interface lisp** *interface-number*
6. **ipv6 pim lisp transport** [**ipv4**]
7. **ip pim sparse-mode**
8. **exit**
9. **interface** *interface-type interface-number*
10. **description** *string*
11. **ip pim sparse-mode**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**distributed**] **Example:** `Device(config)# ip multicast-routing`	Enables IP multicast routing.

	Command or Action	Purpose

Step 4	Enter one of the following:
	• **ip pim rp-address** *rp-address*
	• **ip pim ssm** {**default** | **range** {*access-list-number* | *access-list-name*}}
	Example:
	`Device(config)# ip pim rp-address 10.1.0.2`
	Example:
	`Device(config)# ip pim ssm default`
		• Statically configures the address of a Protocol Independent Multicast (PIM) rendezvous point (RP) for multicast groups.
		• Defines the Source Specific Multicast (SSM) range of IP multicast addresses.

Step 5	**interface lisp** *interface-number* **Example:** `Device(config)# interface LISP0`	Selects a LISP interface to configure and enters interface configuration mode.
Step 6	**ipv6 pim lisp transport** [**ipv4**] **Example:** `Device(config-if)# ipv6 pim lisp transport unicast ipv4`	Selects a LISP interface to configure and enters interface configuration mode.
Step 7	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables Protocol Independent Multicast (PIM) on an interface for sparse-mode operation.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**interface** *interface-type interface-number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Configures the LISP interface facing the site and enters interface configuration mode.
Step 10	**description** *string* **Example:** `Device(config-if)# description Link To Site`	Configures a description text for the interface.
Step 11	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables Protocol Independent Multicast (PIM) on an interface for sparse-mode operation.

	Command or Action	Purpose
Step 12	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Configuring LISP Multicast in VRFs

Perform this task to enable the LISP multicast functionality on an xTR with Virtual Routing and Forwarding (VRF) mode configured.

Before you begin

Ensure that generic multicast functionality has been enabled on the required devices of the LISP site and that PIM sparse mode has been enabled on the required interfaces of these devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **address-family ipv4**
5. **exit**
6. **exit**
7. **ip multicast-routing vrf** *vrf-name* [**distributed**]
8. Enter one of the following:
   - **ip pim vrf** *vrf-name* **rp-address** *ip-address*
   - **ip pim vrf** *vrf-name* **ssm** {**default** | **range** {*access-list-number* | *access-list-name*}}
9. **interface lisp** *interface-number*
10. **ip pim sparse-mode**
11. **exit**
12. **interface** *interface-type interface-number*
13. **vrf forwarding** *vrf-name*
14. **description** *string*
15. **ip pim** *sparse-mode*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition VRF1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 4	**address-family ipv4** **Example:** `Device(config-vrf)# address-family ipv4`	Configures an address family for the VRF and enters VRF address family configuration mode.
Step 5	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 6	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 7	**ip multicast-routing vrf** *vrf-name* [**distributed**] **Example:** `Device(config)# ip multicast-routing vrf VRF1 distributed`	Enables IP multicast routing.

Step 8	Enter one of the following:
	• **ip pim vrf** *vrf-name* **rp-address** *ip-address*
	• **ip pim vrf** *vrf-name* **ssm** {**default** | **range** {*access-list-number* | *access-list-name*}}
	Example:
	`Device(config)# ip pim vrf VRF1 rp-address 10.1.0.2`
	Example:
	`Device(config)# ip pim vrf VRF1 ssm default`
		• Statically configures the address of a Protocol Independent Multicast (PIM) rendezvous point (RP) for multicast groups.
		• Defines the Source Specific Multicast (SSM) range of IP multicast addresses.

Step 9	**interface lisp** *interface-number* **Example:** `Device(config)# interface lisp 22.10`	Selects a LISP interface to configure and enters interface configuration mode.
Step 10	**ip pim sparse-mode** **Example:**	Enables Protocol Independent Multicast (PIM) on an interface for sparse-mode operation.

	Command or Action	Purpose
	`Device(config-if)# ip pim sparse-mode`	
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 12	**interface** *interface-type interface-number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Configures the LISP interface facing the site and enters interface configuration mode.
Step 13	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding VRF1`	Enables VRF forwarding on the interface.
Step 14	**description** *string* **Example:** `Device(config-if)# description Link To Site`	Configures a description text for the interface.
Step 15	**ip pim** *sparse-mode* **Example:** `Device(config-if)# ip pim sparse-mode.`	Enables Protocol Independent Multicast (PIM) on an interface for sparse-mode operation.
Step 16	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Verifying LISP Multicast

Perform this task to verify the configuration of LISP multicast routes on a device.

SUMMARY STEPS

1. **show ip mroute** *multicast-ip-address*
2. **ping** *multicast-ip-address*

DETAILED STEPS

Step 1 **show ip mroute** *multicast-ip-address*

Example:

The following example shows how the IP multicast routing table is displayed using the **show ip mroute** command:

```
Device# show ip mroute 239.4.4.4

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.4.4.4), 00:06:25/00:02:39, RP 10.1.0.2, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 10.1.0.2
  Outgoing interface list:
    Loopback2, Forward/Sparse, 00:06:24/00:02:39

(*, 224.0.1.40), 00:06:25/00:02:37, RP 10.1.0.2, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 10.1.0.2
  Outgoing interface list:
    Loopback2, Forward/Sparse, 00:06:24/00:02:37
```

Step 2 **ping** *multicast-ip-address*

Example:

The following example shows how to verify basic multicast network connectivity by pinging the multicast address:

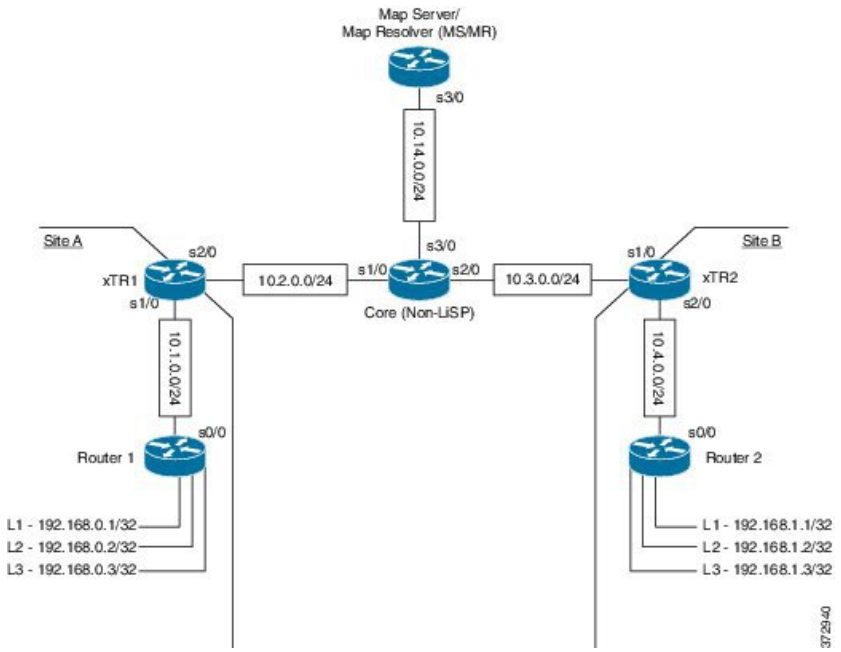
```
Device# ping 239.4.4.4

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.4.4.4, timeout is 2 seconds:

Reply to request 0 from 192.168.0.1, 15 ms
Reply to request 0 from 10.1.0.2, 58 ms
Reply to request 0 from 10.1.0.2, 58 ms
Reply to request 0 from 10.1.0.1, 35 ms
Reply to request 0 from 10.1.0.2, 34 ms
Reply to request 0 from 10.1.0.1, 15 ms
```

Configuration Examples for LISP Multicast

Example: Configuring LISP Multicast

The following example shows how to configure LISP Multicast in the topology given below:

Figure 167: LISP Multicast Topology



Router 1

The following example shows how to configure LISP multicast in Router 1:

```
Device# show startup-config


!
ip multicast-routing
!
interface Loopback1
 ip address 192.168.0.1 255.255.255.255
 ip pim sparse-mode
 ip igmp join-group 239.4.4.4
 serial restart-delay 0
!
interface Loopback2
 ip address 192.168.0.2 255.255.255.255
 ip pim sparse-mode
 ip igmp join-group 239.4.4.4
 serial restart-delay 0
!
interface Loopback3
 ip address 192.168.0.3 255.255.255.255
```

```
 ip pim sparse-mode
 ip igmp join-group 239.4.4.4
 serial restart-delay 0
!
interface Serial0/0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-mode
 serial restart-delay 0
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.0.0
 default-information originate
!
ip forward-protocol nd
!
ip pim rp-address 10.1.0.2
!
!
End
```

The following example shows how to verify the configuration of LISP multicast routes in Router 1:

```
Device# show ip mroute


IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.4.4.4), 00:00:49/00:02:16, RP 10.1.0.2, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 10.1.0.2
  Outgoing interface list:
    Loopback2, Forward/Sparse, 00:00:48/00:02:12

(*, 224.0.1.40), 00:00:49/00:02:11, RP 10.1.0.2, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 10.1.0.2
  Outgoing interface list:
    Loopback2, Forward/Sparse, 00:00:48/00:02:11
```

The following example shows how to verify basic multicast network connectivity from Router 1 by pinging the multicast address:

```
Device# ping 239.4.4.4
```

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.4.4.4, timeout is 2 seconds:

Reply to request 0 from 192.168.0.1, 9 ms
Reply to request 0 from 10.1.0.2, 48 ms
Reply to request 0 from 192.168.0.2, 16 ms
Reply to request 0 from 192.168.0.3, 16 ms
Reply to request 0 from 10.1.0.1, 38 ms
Reply to request 0 from 10.1.0.2, 38 ms
Reply to request 0 from 10.1.0.2, 29 ms
Reply to request 0 from 10.1.0.1, 9 ms
```

xTR1

The following example shows how to configure LISP multicast in xTR1:

```
Device# show startup-config


!
ip multicast-routing
!
interface LISP0
 ip pim sparse-mode
!
interface Serial1/0
 ip address 10.1.0.2 255.255.255.0
 ip pim sparse-mode
 serial restart-delay 0
!
interface Serial2/0
 ip address 10.2.0.1 255.255.255.0
 serial restart-delay 0
!
router lisp
 database-mapping 192.168.0.0/24 10.2.0.1 priority 1 weight 100
 ipv4 itr map-resolver 10.14.0.14
 ipv4 itr
 ipv4 etr map-server 10.14.0.14 key password123
 ipv4 etr
 exit
!
!
router rip
 version 2
 network 10.0.0.0
 default-information originate
!
ip pim rp-address 10.1.0.2
ip route 0.0.0.0 0.0.0.0 10.2.0.2
!
```

Router 2

The following example shows how to configure LISP multicast in Router 2:

```
Device# show startup-config


!
ip multicast-routing
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.255
 ip pim sparse-mode
 ip igmp join-group 239.4.4.4
 serial restart-delay 0
!
interface Loopback2
 ip address 192.168.1.2 255.255.255.255
 ip pim sparse-mode
 ip igmp join-group 239.4.4.4
 serial restart-delay 0
!
interface Loopback3
 ip address 192.168.1.3 255.255.255.255
 ip pim sparse-mode
 ip igmp join-group 239.4.4.4
 serial restart-delay 0
!
interface Serial0/0
 ip address 10.4.0.2 255.255.255.0
 ip pim sparse-mode
 serial restart-delay 0
!
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 default-information originate
!
ip forward-protocol nd
!
!
ip pim rp-address 10.1.0.2
!
!
End
```

The following example shows how to verify the configuration of LISP multicast routes in Router 2:

```
Device# show ip mroute


IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group
```

```
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.4.4.4), 00:12:59/00:02:01, RP 10.4.0.1, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 10.4.0.1
  Outgoing interface list:
    Loopback2, Forward/Sparse, 00:12:58/00:02:01

(*, 224.0.1.40), 00:12:59/00:02:03, RP 10.4.0.1, flags: SJCL
  Incoming interface: Serial0/0, RPF nbr 10.4.0.1
  Outgoing interface list:
    Loopback2, Forward/Sparse, 00:12:58/00:02:03
```

The following example shows how to verify basic multicast network connectivity from Router 2 by pinging the multicast address:

```
Device# ping 239.4.4.4


Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.4.4.4, timeout is 2 seconds:

Reply to request 0 from 192.168.1.1, 2 ms
Reply to request 0 from 10.3.0.2, 26 ms
Reply to request 0 from 10.4.0.1, 26 ms
Reply to request 0 from 192.168.1.2, 2 ms
Reply to request 0 from 192.168.1.3, 8 ms
Reply to request 0 from 10.4.0.1, 16 ms
Reply to request 0 from 10.4.0.1, 16 ms
Reply to request 0 from 10.4.0.2, 2 ms
```

xTR2

The following example shows how to configure LISP multicast in xTR2:

```
Device# show startup-config


!
ip multicast-routing
!
interface LISP0
 ip pim sparse-mode
!
!
interface Serial1/0
 ip address 10.3.0.2 255.255.255.0
 serial restart-delay 0
!
interface Serial2/0
 ip address 10.4.0.1 255.255.255.0
 ip pim sparse-mode
 serial restart-delay 0
!
!
router lisp
 database-mapping 192.168.1.0/24 10.3.0.2 priority 1 weight 100
```

```
 ipv4 itr map-resolver 10.14.0.14
 ipv4 itr
 ipv4 etr map-server 10.14.0.14 key Amel
 ipv4 etr
 exit
!
router rip
 version 2
 network 10.0.0.0
 default-information originate
!
ip pim rp-address 10.1.0.2
ip route 0.0.0.0 0.0.0.0 10.3.0.1
!
```

MS/MR

The following example shows how to configure LISP multicast in MS/MR:

```
Device# show startup-config


!
ip multicast-routing
!
interface Serial3/0
 ip address 10.14.0.14 255.255.255.0
 serial restart-delay 0
!
!
router lisp
 site Site-A
  authentication-key password123
  eid-prefix 192.168.0.0/24
  exit
 !
 site Site-B
  authentication-key Amel
  eid-prefix 192.168.1.0/24
  exit
 !
 ipv4 map-server
 ipv4 map-resolver
 exit
!
ip route 0.0.0.0 0.0.0.0 10.14.0.1
!
```

Core

The following example shows how to configure LISP multicast in the Core router:

```
Device# show startup-config


!
ip multicast-routing
```

```
!
interface Ethernet0/0
 ip address 10.14.0.1 255.255.255.0
 serial restart-delay 0
!
interface Serial1/0
 ip address 10.2.0.2 255.255.255.0
 serial restart-delay 0
!
interface Serial2/0
 ip address 10.3.0.1 255.255.255.0
 serial restart-delay 0
!
```

Example: Configuring LISP Multicast in VRFs

The following example shows how to enable and configure a simple LISP site with one IPv4 Routing locator (RLOC) and one IPv4 Endpoint identifier (EID) using xTR, a device which functions both as an Ingress tunnel router (ITR) and an Egress tunnel router (ETR), functionality and using a LISP map server and map resolver for mapping services:

```
Device> enable
Device# configure terminal
Device(config)# vrf definition VRF1
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# ip multicast-routing vrf VRF1 [distributed]
Device(config)# ip pim vrf VRF1 ssm range LIST1
Device(config)# router lisp 22
Device(config-router-lisp)# eid-table vrf VRF1 instance-id 10
Device(config-router-lisp-eid-table)# database-mapping 198.51.100.0/24 192.0.2.10 priority
 1 weight 100
Device(config-router-lisp-eid-table)# exit
Device(config-router-lisp)# ipv4 itr
Device(config-router-lisp)# ipv4 etr
Device(config-router-lisp)# ipv4 itr map-resolver 192.0.2.10
Device(config-router-lisp)# ipv4 etr map-server 192.0.2.10 key 0 some-key
Device(config-router-lisp)# exit
Device(config)# interface lisp 22.10
Device(config-if)# ip pim sparse-mode
Device(config-if)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 192.0.2.20
Device(config)# end
```

Additional References for LISP Multicast

The following sections provide references related to the Locator ID Separation Protocol.

Related Documents

Document Title	Location
Cisco IOS commands	Cisco IOS Master Command List, All Releases
LISP commands	Cisco IOS IP Routing: LISP Command Reference

Standards

Standard	Title
Address family numbers	IANA Address Family Numbers

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 6830	Locator/ID Separation Protocol (LISP) http://tools.ietf.org/html/
RFC 6831	LISP Multicast http://tools.ietf.org/html/rfc6831
RFC 6832	Interworking LISP and Non-LISP Sites http://tools.ietf.org/html/rfc6832
RFC 6833	LISP Map Server Interface http://tools.ietf.org/html/rfc6833
RFC 6834	LISP Map-Versioning http://tools.ietf.org/html/rfc6834
RFC 6835	LISP Internet Groper http://tools.ietf.org/html/rfc6835
RFC 6836	LISP Alternative Topology (LISP+ALT) http://tools.ietf.org/html/rfc6836

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Multicast

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 193: Feature Information for LISP Multicast

Feature Name	Releases	Feature Information
LISP Multicast		The LISP Multicast feature introduces support for carrying multicast traffic over a Locator ID Separation Protocol (LISP) overlay and allows source multicast sites and receiver multicast sites to send and receive multicast packets over a unicast RLOC core.

CHAPTER 180

LISP Shared Model Virtualization

This guide describes how to configure Locator ID Separation Protocol (LISP) shared model virtualization using IOS XE Software on all LISP-related devices, including the Egress Tunnel Router, Ingress Tunnel Router (ITR), Proxy ETR (PETR), Proxy ITR (PITR), Map Resolver (MR), and Map Server (MS).

LISP implements a new routing architecture that utilizes a "level of indirection" to separate an IP address into two namespaces: Endpoint Identifiers (EIDs), which are assigned to end-hosts, and Routing Locators (RLOCs), which are assigned to devices (primarily routers) that make up the global routing system. Splitting EID and RLOC functions yields several advantages including: improved routing system scalability, multihoming with ingress traffic engineering; efficient IPv6 Transition support; high-scale virtualization/multitenancy support; data center/VM-mobility support, including session persistence across mobility events; and seamless mobile node support.

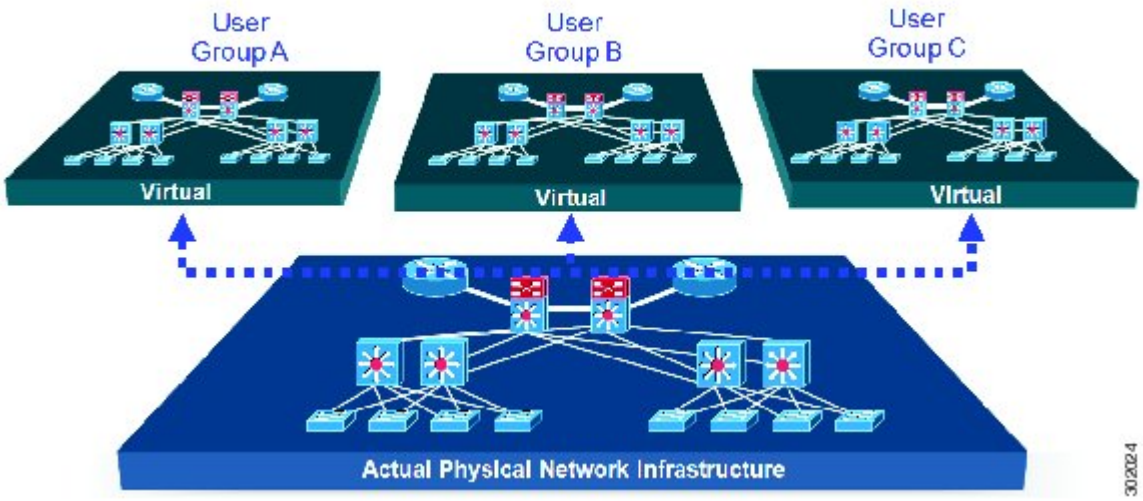
Information About LISP Shared Model Virtualization

Overview of LISP Virtualization

Deploying physical network infrastructure requires both capital investments for hardware, as well as manpower investments for installation and operational management support. When distinct user groups within an organization desire to control their own networks, it rarely makes economic sense for these user groups to deploy and manage separate physical networks. Physical plants are rarely utilized to their fullest, resulting in stranded capacity (bandwidth, processor, memory, etc.). In addition, the power, rack space, and cooling needs to physical plants do not satisfy modern “green” requirements. Network virtualization offers the opportunity to satisfy organizational needs, while efficiently utilizing physical assets.

The purpose of network virtualization, as shown in the figure below, is to create multiple, logically separated topologies across one common physical infrastructure.

Figure 168: LISP Deployment Environment



When considering the deployment of a virtualized network environment, take into account both the device and the path level.

Device Level Virtualization

Virtualization at the device level entails the use of the virtual routing and forwarding (VRF) to create multiple instances of Layer 3 routing tables, as illustrated in the figure below. VRFs provide segmentation across IP addresses, allowing for overlapped address space and traffic separation. Separate routing, QoS, security, and management policies can be applied to each VRF instance. An IGP or EGP routing process is typically enabled within a VFR, just as it would be in the global (default) routing table. As described in detail below, LISP binds VRFs to instance IDs for similar purposes.

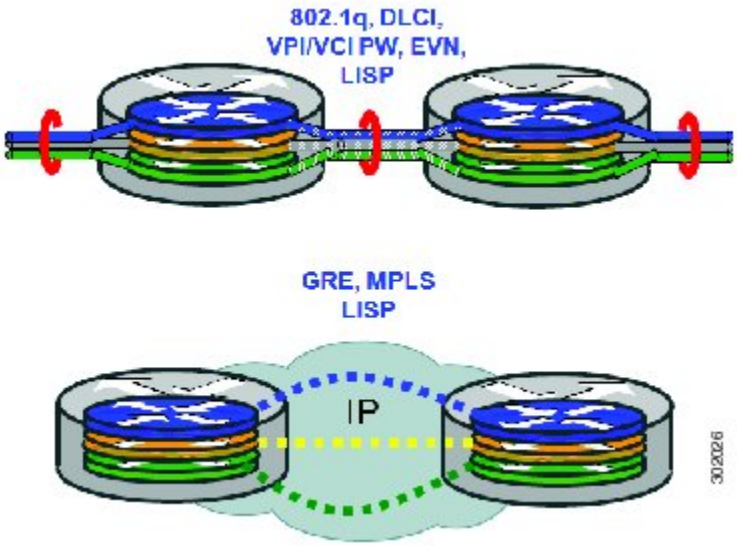
Figure 169: Device Level Virtualization

Path Level Virtualization

VRF table separation is maintained across network paths using any number of traditional mechanisms, as illustrated in the figure below. Single-hop path segmentation (hop-by-hop) is typically accomplished by techniques such as 802.1q VLANs, VPI/VCI PW, or EVN. LISP can also be used. Traditional multi-hop

mechanisms include MPLS and GRE tunnels. As described in detail below, LISP binds VRFs to instance IDs (IIDs), and then these IIDs are included in the LISP header to provide data plane (traffic flow) separation for single or multihop needs.

Figure 170: Path Level Virtualization



LISP Virtualization at the Device Level

Recalling that LISP implements Locator ID separation and, in so doing, creates two namespaces (EIDs and RLOCs), it is easy to see that LISP virtualization can consider both EID and RLOC namespaces for virtualization. That is, either or both can be virtualized.

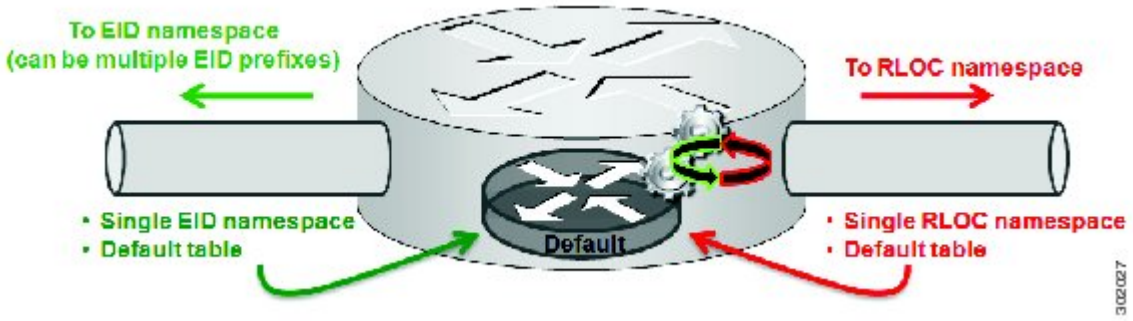
- EID virtualization—Enabled by binding a LISP instance ID to an EID VRF. Instance IDs are numerical tags defined in the LISP canonical address format (LCAF) draft, and are used to maintain address space segmentation in both the control plane and data plane.
- RLOC virtualization—Tying locator addresses and associated mapping services to the specific VRF within which they are reachable enables RLOC virtualization.

Because LISP considers virtualization of both EID and RLOC namespaces, two models of operation are defined: shared model and parallel model. For completeness, the discussions below begin first with a review of the default (non-virtualized) model of LISP, and then cover the details of shared and parallel models.

Default (Non-Virtualized) LISP Model

By default, LISP is not virtualized in either EID space or RLOC space. That is, unless otherwise configured, both EID and RLOC addresses are resolved in the default (global) routing table. This concept is illustrated in the figure below.

Figure 171: Default (Non-Virtualized) LISP Model (Resolves Both EID and RLOC Addresses in the Default (Global) Routing Table.

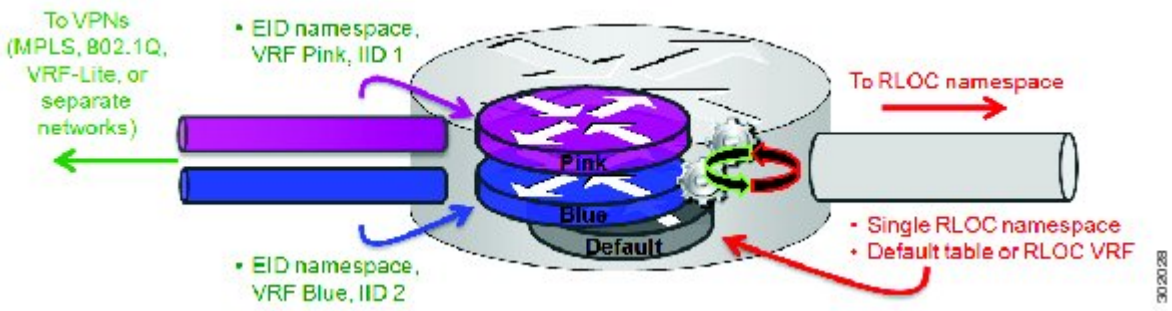


As shown in the figure above, both EID and RLOC addresses are resolved in the default table. The mapping system must also be reachable via the default table. This default model can be thought of as a single instantiation of the parallel model of LISP virtualization where EID and RLOC addresses are within the same namespace such as is the case in this default table.

LISP Shared Model Virtualization

LISP shared model virtualized EID space is created by binding VRFs associated with an EID space to Instance IDs. A common, shared locator space is used by all virtualized EIDs. This concept is illustrated in the figure below.

Figure 172: LISP shared model virtualization resolves EIDs within VRFs tied to Instance IDs. RLOC addresses are resolved in a common (shared) address space. The default (global) routing table is shown as the shared space.

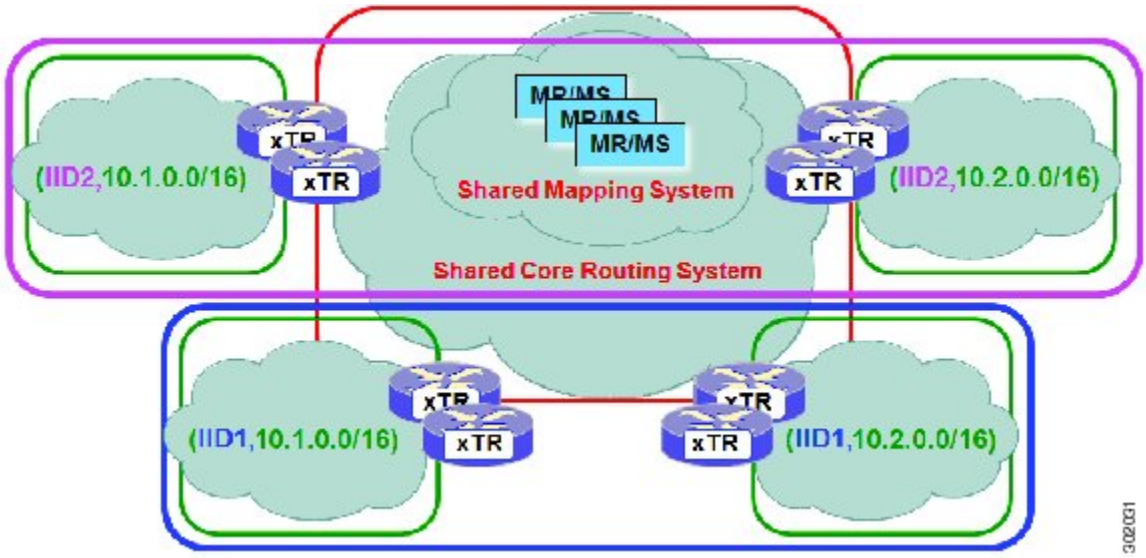


As shown in the figure above, EID space is virtualized through its association with VRFs, and these VRFs are tied to LISP Instance IDs to segment the control plane and data plane in LISP. A common, shared locator space, the default (global) table as shown in the figure above, is used to resolve RLOC addresses for all virtualized EIDs. The mapping system must also be reachable via the common locator space.

LISP Shared Model Virtualization Architecture

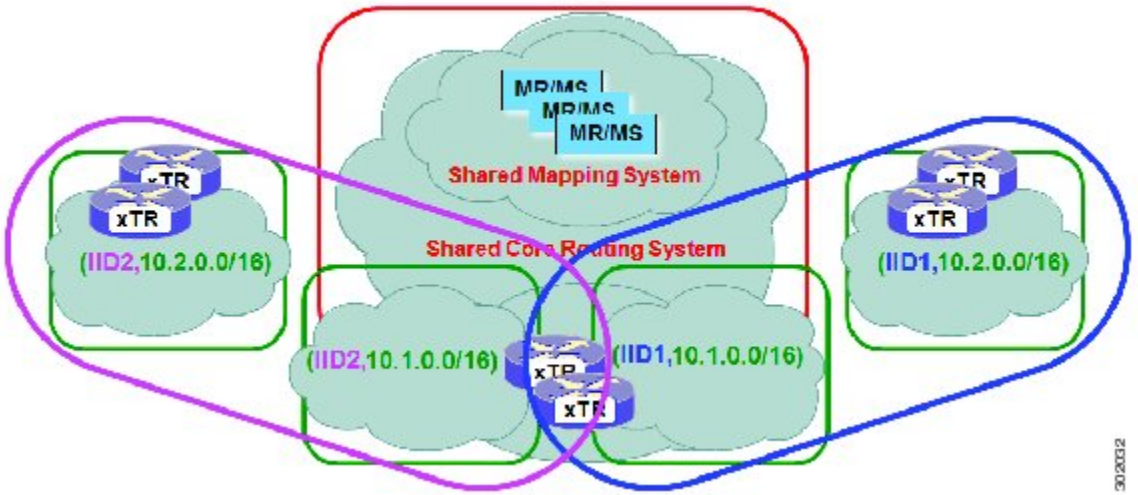
Architecturally, LISP shared model virtualization can be deployed in single or multitenancy configurations. In the shared model single tenancy case, xTRs are dedicated to a customer but share infrastructure with other customers. Each customer and all sites associated with it use the same instance ID and are part of a VPN using their own EID namespace as shown in the figure below.

Figure 173: In a LISP shared model single tenancy use case, customers use their own xTRs and a shared common core network and mapping system. LISP instance IDs segment the LISP data plane and control plane.



In the shared model multitenancy case, a set of xTRs is shared (virtualized) among multiple customers. These customers also share a common infrastructure with other single and multitenant customers. Each customer and all sites associated with it use the same instance ID and are part of a VPN using their own EID namespace as shown in the figure below.

Figure 174: In a LISP shared model multitenancy use case, customer's use shared xTRs and a shared common core network and mapping system. LISP instance IDs segment the LISP data plane and control plane.



LISP Shared Model Virtualization Implementation Considerations and Caveats

When LISP Shared Model is implemented, several important considerations and caveats are important. Instance IDs must be unique to an EID VRF. Review the example below:

```
xTR-1(config)# vrf definition alpha
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config)# vrf definition beta
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config-vrf)# exit
xTR-1(config)# router lisp
xTR-1(config-router-lisp)# eid-table vrf alpha instance-id 101
xTR-1(config-router-lisp-eid-table)# exit
xTR-1(config-router-lisp)# eid-table vrf beta instance-id 101
Instance ID 101 is bound to the vrf alpha EID table.
```

In the above example, two EID VRFs are created: alpha and beta. Under the **router lisp** command, an EID table VRF named alpha is specified and associated with the instance ID 101. Next, an EID table VRF named beta is specified and also associated with the instance ID 101. As indicated by the router, this is not permissible since instance ID 101 is already associated with the EID VRF named alpha. That is, you cannot connect the same instance-id to more than one EID VRF.

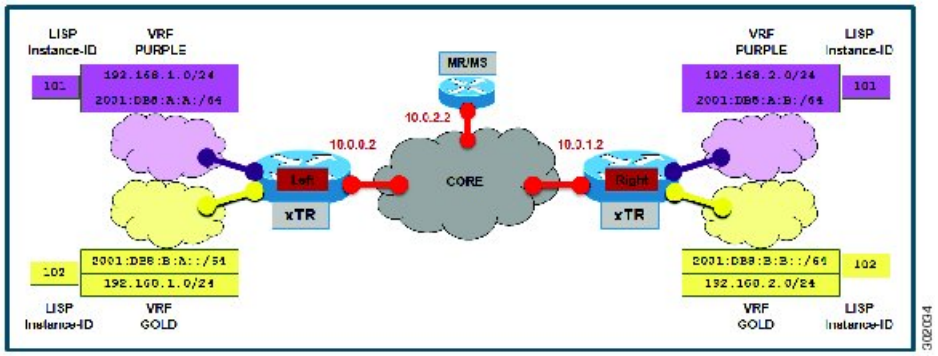
How to Configure LISP Shared Model Virtualization

Configure Simple LISP Shared Model Virtualization

Perform this task to enable and configure LISP ITR/ETR (xTR) functionality with LISP map server and map resolver to implement LISP shared model virtualization. This LISP shared model reference configuration is for a very simple two-site LISP topology, including xTRs and an MS/MR.

The configuration implemented in this task and illustrated in the figure below shows a basic LISP shared model virtualization solution. In this example, two LISP sites are deployed, each containing two VRFs: PURPLE and GOLD. LISP is used to provide virtualized connectivity between these two sites across a common IPv4 core, while maintaining address separation between the two VRFs.

Figure 175: Simple LISP Site with virtualized IPv4 and IPv6 EIDs and a shared IPv4 core



Each LISP Site uses a single edge router configured as both an ITR and ETR (xTR), with a single connection to its upstream provider. The RLOC is IPv4, and IPv4 and IPv6 EID prefixes are configured. Each LISP site registers to a map server/map resolver (MS/MR) device located in the network core within the shared RLOC address space. The topology used in this most basic LISP configuration is shown in the figure above.

The components illustrated in the topology shown in the figure above are described below:

- **LISP site:**

- The CPE functions as a LISP ITR and ETR (xTR).
- Both LISP xTRs have two VRFs: GOLD and PURPLE, with each VRF containing both IPv4 and IPv6 EID-prefixes, as shown in the figure above. Note the overlapping prefixes, used for illustration purposes. A LISP instance-id is used to maintain separation between two VRFs. Note that in this example, the share key is configured "per-site" and not "per-VRF." (Case 2 illustrates a configuration where the shared key is per-VPN.)
- Each LISP xTR has a single RLOC connection to a shared IPv4 core network.

- **Mapping system:**
  - One map server/map resolver system is shown in the figure above and assumed available for the LISP xTR to register to. The MS/MR has an IPv4 RLOC address of 10.0.2.2, within the shared IPv4 core.
  - The map server site configurations are virtualized using LISP instance-ids to maintain separation between the two VRFs.

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map-server and map-resolver for mapping services. The example configurations at the end of this task show the full configuration for two xTRs (xTR1 and xTR2).

Before you begin

The configuration below assumes that the referenced VRFs were created using the **vrf definition** command.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **eid-table vrf***vrf-name* **instance-id** *instance-id*
4. Do one of the following:
   - **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
   - **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
5. Repeat Step 4 until all EID-to-RLOC mappings for the LISP site are configured.
6. **exit**
7. **ipv4 itr**
8. **ipv4 etr**
9. **ipv4 itr map-resolver** *map-resolver-address*
10. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
11. **ipv6 itr**
12. **ipv6 etr**
13. **ipv6 itr map-resolver** *map-resolver-address*
14. **ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key*
15. **exit**
16. **ip route** *ipv4-prefix next-hop*
17. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (Cisco IOS XE software only).
Step 3	**eid-table vrf***vrf-name* **instance-id** *instance-id* **Example:** `Router(config-router-lisp)# eid-table vrf GOLD instance-id 102`	Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode. • In this example, the VRF table GOLD and instance-id 102 are associated together.

Step 4	Do one of the following:
	• **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
	• **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
	Example:
	`Router(config-router-lisp-eid-table)# database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 100`
	Example:
	`Router(config-router-lisp-eid-table)# database-mapping 192.168.1.0/24 ipv4-interface Ethernet0/0 priority 1 weight 100`
		Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
		• In the first example, a single IPv4 EID prefix, 192.168.1.0/24, is being associated with the single IPv4 RLOC 10.0.0.2.
		• In the second example, the alternative configuration shows the use of the dynamic interface form of the **database-mapping** command. This form is useful when the RLOC address is obtained dynamically, such as via DHCP.

Step 5	Repeat Step 4 until all EID-to-RLOC mappings for the LISP site are configured. **Example:** `Router(config-router-lisp-eid-table)# database-mapping 2001:db8:b:a::/64 10.0.0.2 priority 1 weight 100`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
Step 6	**exit** **Example:** `Router(config-router-lisp-eid-table)# exit`	Exits eid-table configuration submode and returns to LISP configuration mode.

	Command or Action	Purpose
Step 7	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 8	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 9	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 10	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.0.2.2 key 0 Left-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • The map server must be configured with EID prefixes and instance IDs matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 11	**ipv6 itr** **Example:** `Router(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 12	**ipv6 etr** **Example:**	Enables LISP ETR functionality for the IPv6 address family.

	Command or Action	Purpose
	`Router(config-router-lisp)# ipv6 etr`	
Step 13	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv6 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-resolver is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 14	**ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.0.2.2 key 0 Left-key`	Configures a locator address for the LISP map-server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system. • The map-server must be configured with EID prefixes and instance IDs matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map-server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 15	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 16	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP

	Command or Action	Purpose
		EID and the destination matches one of the following entries: • a current map-cache entry • a default route with a legitimate next-hop • no route at all In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.
Step 17	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

The examples below show the complete configuration for the LISP topology illustrated in the figure shown above the task steps and follows the examples in the steps in this task. On the xTRs, the VRFs and EID prefixes are assumed to be attached to VLANs configured on the devices.

Example configuration for the Left xTR:

```
hostname Left-xTR
!
ipv6 unicast-routing
!
vrf definition PURPLE
 address-family ipv4
 exit
 address-family ipv6
 exit
!
vrf definition GOLD
 address-family ipv4
 exit
 address-family ipv6
 exit
!
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0
!
interface Ethernet1/0.1
 encapsulation dot1q 101
 vrf forwarding PURPLE
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:A:A::1/64
!
interface Ethernet1/0.2
 encapsulation dot1q 102
 vrf forwarding GOLD
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:B:A::1/64
!
```

```
router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:A:A::/64 10.0.0.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:B:A::/64 10.0.0.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Left-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Left-key
 ipv6 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 10.0.0.1
!
```

Example configuration for Right xTR:

```
hostname Right-xTR
!
ipv6 unicast-routing
!
vrf definition PURPLE
 address-family ipv4
 exit
 address-family ipv6
 exit
!
vrf definition GOLD
 address-family ipv4
 exit
 address-family ipv6
 exit
!
interface Ethernet0/0
 ip address 10.0.1.2 255.255.255.0
!
interface Ethernet1/0.1
 encapsulation dot1q 101
 vrf forwarding PURPLE
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:A:B::1/64
!
interface Ethernet1/0.2
 encapsulation dot1q 102
 vrf forwarding GOLD
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:B:B::1/64
!
router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 1
  database-mapping 2001:DB8:A:B::/64 10.0.1.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 1
  database-mapping 2001:DB8:B:B::/64 10.0.1.2 priority 1 weight 1
 exit
 !
```

```
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Right-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Right-key
 ipv6 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 10.0.1.1
!
```

Configuring a Private LISP Mapping System for LISP Shared Model Virtualization

Perform this task to configure and enable standalone LISP map server/map resolver functionality for LISP shared model virtualization. In this task, a Cisco router is configured as a standalone map server/map resolver (MR/MS) for a private LISP mapping system. Because the MR/MS is configured as a stand-alone device, it has no need for LISP Alternate Logical Topology (ALT) connectivity. All relevant LISP sites must be configured to register with this map server so that this map server has full knowledge of all registered EID Prefixes within the (assumed) private LISP system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **site** *site-name*
5. **authentication-key** [*key-type*] *authentication-key*
6. **eid-prefix instance-id** *instance-id EID-prefix*
7. **eid-prefix instance-id** *instance-id EID-prefix*
8. **exit**
9. **ipv4 map-resolver**
10. **ipv4 map-server**
11. **ipv6 map-resolver**
12. **ipv6 map-server**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS only).
Step 4	**site** *site-name* **Example:** `Router(config-router-lisp)# site Left`	Specifies a LISP site named Left and enters LISP site configuration mode. **Note** A LISP site name is locally significant to the map server on which it is configured. It has no relevance anywhere else. This name is used solely as an administrative means of associating EID-prefix or prefixes with an authentication key and other site-related mechanisms.
Step 5	**authentication-key** [*key-type*] *authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 Left-key`	Configures the password used to create the SHA-2 HMAC hash for authenticating the map register messages sent by an ETR when registering to the map server. **Note** The LISP ETR must be configured with an identical authentication key as well as matching EID prefixes and instance IDs.
Step 6	**eid-prefix instance-id** *instance-id EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 102 192.168.1.0/24`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. Repeat this step as necessary to configure additional EID prefixes under this LISP site. • In this example, the IPv4 EID prefix 192.168.1.0/24 and instance ID 102 are associated together. To complete this task, an IPv6 EID prefix is required.
Step 7	**eid-prefix instance-id** *instance-id EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 102 2001:db8:a:b::/64`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. • In this example, the IPv6 EID prefix 2001:db8:a:b::/64 and instance ID 102 are associated together.
Step 8	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 9	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.

	Command or Action	Purpose
Step 10	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 11	**ipv6 map-resolver** **Example:** `Router(config-router-lisp)# ipv6 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family.
Step 12	**ipv6 map-server** **Example:** `Router(config-router-lisp)# ipv6 map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family.
Step 13	**end** **Example:** `Router(config-router-lisp)# end`	Exits LISP configuration mode and returns to privileged EXEC mode.

Example:

Example configuration for the map server/map resolver.

```
hostname MSMR
!
interface Ethernet0/0
 ip address 10.0.2.2 255.255.255.0
!
 router lisp
  !
  site Left
   authentication-key Left-key
   eid-prefix instance-id 101 192.168.1.0/24
   eid-prefix instance-id 101 2001:DB8:A:A::/64
   eid-prefix instance-id 102 192.168.1.0/24
   eid-prefix instance-id 102 2001:DB8:B:A::/64
   exit
  !
  site Right
   authentication-key Right-key
   eid-prefix instance-id 101 192.168.2.0/24
   eid-prefix instance-id 101 2001:DB8:A:B::/64
   eid-prefix instance-id 102 192.168.2.0/24
   eid-prefix instance-id 102 2001:DB8:B:B::/64
   exit
  !
  ipv4 map-server
  ipv4 map-resolver
  ipv6 map-server
  ipv6 map-resolver
  exit
```

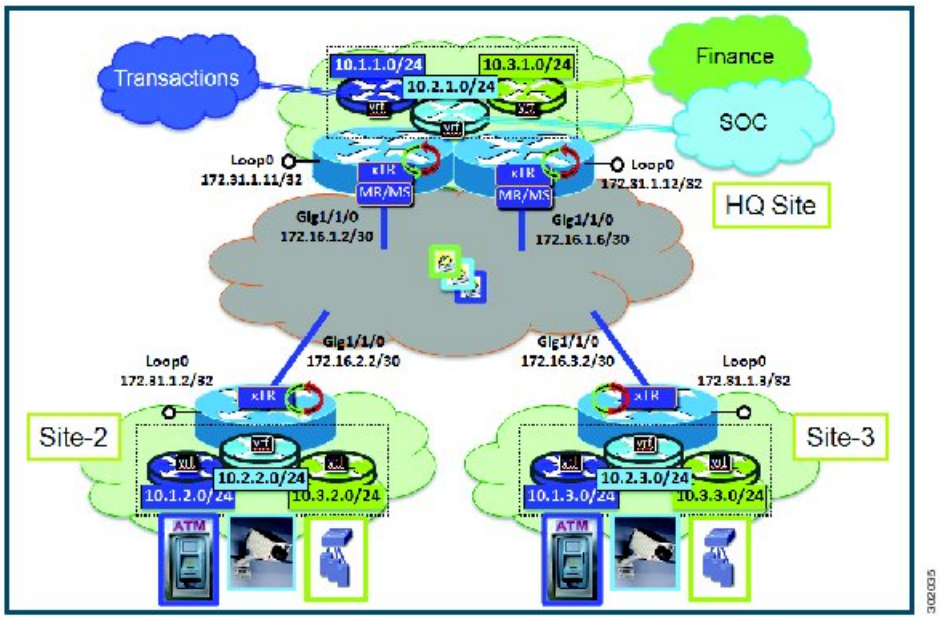
```
!
ip route 0.0.0.0 0.0.0.0 10.0.2.1
```

Configure Large-Scale LISP Shared Model Virtualization

Perform this task to enable and configure LISP ITR/ETR (xTR) functionality with LISP map server and map resolver to implement LISP shared model virtualization. This LISP shared model reference configuration is for a large-scale, multiple-site LISP topology, including xTRs and multiple MS/MRs.

The configuration demonstrated in this task shows a more complex, larger scale LISP virtualization solution. In this task, an enterprise is deploying LISP Shared Model where EID space is virtualized over a shared, common core network. A subset of their entire network is illustrated in Figure 12. In this figure, three sites are shown: a multihomed "Headquarters" (HQ) site, and two remote office sites. The HQ site routers are deployed as xTRs and also as map resolver/map servers. The remote site routers only act as xTRs, and use the MS/MRs at the HQ site for LISP control plane support.

Figure 176: Large Scale LISP Site with Virtualized IPv4 EIDs and a Shared IPv4 Core



The components illustrated in the topology shown in the figure above are described below:

- **LISP site:**
  - Each CPE router functions as a LISP ITR and ETR (xTR), as well as a Map-Server/Map-Resolver (MS/MR).
  - Both LISP xTRs have three VRFs: TRANS (for transactions), SOC (for security operations), and FIN (for financials). Each VRF contains only IPv4 EID-prefixes. Note that no overlapping prefixes are used, but segmentation between each VRF by LISP instance-ids makes this possible. Also note that in this example, the separate authentication key is configured “per-vrf﷒? and not “per-site.﷒? This affects both the xTR and MS configurations.
  - The HQ LISP Site is multi-homed to the shared IPv4 core, but each xTR at the HQ site has a single RLOC.

- Each CPE also functions as an MS/MR to which the HQ and Remote LISP sites can register.
- The map server site configurations are virtualized using LISP instance IDs to maintain separation between the three VRFs.

- **LISP remote sites:**
  - Each remote site CPE router functions as a LISP ITR and ETR (xTR).
  - Each LISP xTRs has the same three VRFs as the HQ Site: TRANS, SOC, and FIN. Each VRF contains only IPv4 EID-prefixes.
  - Each remote site LISP xTR has a single RLOC connection to a shared IPv4 core network.

Before you begin

The configuration below assumes that the referenced VRFs were created using the **vrf definition** command.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **site** *site-name*
4. **authentication-key** [*key-type*] *authentication-key*
5. **eid-prefix instance-id** *instance-id EID-prefix/prefix-length* **accept-more-specifics**
6. **exit**
7. Repeat steps 3 through 6 for each LISP site to be configured.
8. **ipv4 map-resolver**
9. **ipv4 map-server**
10. **eid-table vrf***vrf-name* **instance-id** *instance-id*
11. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
12. Repeat Step 11 until all EID-to-RLOC mappings within this eid-table vrf and instance ID for the LISP site are configured.
13. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
14. Repeat Step 13 to configure another locator address for the same LISP map server
15. **exit**
16. **ipv4 itr map-resolver** *map-resolver-address*
17. Repeat Step 16 to configure another locator address for the LISP map resolver
18. **ipv4 itr**
19. **ipv4 etr**
20. **exit**
21. **ip route** *ipv4-prefix next-hop*
22. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 3	**site** *site-name* **Example:** `Router(config-router-lisp)# site TRANS`	Specifies a LISP site named TRANS and enters LISP site configuration mode. **Note** A LISP site name is locally significant to the map server on which it is configured. It has no relevance anywhere else. This name is used solely as an administrative means of associating EID-prefix or prefixes with an authentication key and other site-related mechanisms.
Step 4	**authentication-key** [*key-type*] *authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 TRANS-key`	Configures the password used to create the SHA-2 HMAC hash for authenticating the map register messages sent by an ETR when registering to the map server. **Note** The LISP ETR must be configured with an identical authentication key as well as matching EID prefixes and instance IDs.
Step 5	**eid-prefix instance-id** *instance-id EID-prefix/prefix-length* **accept-more-specifics** **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 1 10.1.0.0/16 accept-more-specifics`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. Repeat this step as necessary to configure additional EID prefixes under this LISP site. • In the example, EID-prefix 10.1.0.0/16 and instance-id 1 are associated together. The EID-prefix 10.1.0.0/16 is assumed to be an aggregate covering all TRANS EID-prefixes at all LISP Sites. The keyword **accept-more-specifics** is needed in this case to allow each site to register its more-specific EID-prefix contained within that aggregate. If aggregation is not possible, simply enter all EID-prefixes integrated within instance-id 1.
Step 6	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.

	Command or Action	Purpose
Step 7	Repeat steps 3 through 6 for each LISP site to be configured.	In this example, steps 3 through 6 would be repeated for the site SOC and FIN as illustrated in the complete configuration example at the end of this task.
Step 8	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 9	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 10	**eid-table vrf***vrf-name* **instance-id** *instance-id* **Example:** `Router(config-router-lisp)# eid-table vrf TRANS instance-id 1`	Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode. • In this example, the VRF table TRANS and instance-id 1 are associated together.
Step 11	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp-eid-table)# database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 100`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this example, the EID prefix 10.1.1.0/24 within instance-id 1 at this site is associated with the local IPv4 RLOC 172.16.1.2, as well as with the neighbor xTR RLOC 172.6.1.6.
Step 12	Repeat Step 11 until all EID-to-RLOC mappings within this eid-table vrf and instance ID for the LISP site are configured. **Example:** `Router(config-router-lisp-eid-table)# database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 100`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
Step 13	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.2 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, the map server and authentication-key are specified here, within the eid-table subcommand mode, so that the authentication key is associated only with this instance ID, within this VPN.

	Command or Action	Purpose
		Note The map server must be configured with EID prefixes and instance-ids matching the one(s) configured on this ETR, as well as an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 14	Repeat Step 13 to configure another locator address for the same LISP map server **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.6 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, a redundant map server is configured. (Because the MS is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for registration (and its neighbor xTR/MS/MR at the same site).
Step 15	**exit** **Example:** `Router(config-router-lisp-eid-table)# exit`	Exits eid-table configuration submode and returns to LISP configuration mode.
Step 16	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.2`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • In this example, the map resolver is specified within router lisp configuration mode and inherited into all eid-table instances since nothing is related to any single instance ID. In addition, redundant map resolvers are configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site). • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.)

	Command or Action	Purpose
		Note Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 17	Repeat Step 16 to configure another locator address for the LISP map resolver **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.6`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • In this example, a redundant map resolver is configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site). • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 18	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 19	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 20	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 21	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 172.16.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP

	Command or Action	Purpose
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries: • a current map-cache entry • a default route with a legitimate next-hop • no route at all In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.
Step 22	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

The examples below show the complete configuration for the HQ-RTR-1 and HQ-RTR-2 (xTR/MS/MR located at the HQ Site), and Site2-xTR LISP devices illustrated in the figure above and in this task. Note that both HQ-RTR-1 and HQ-RTR-2 are provided in order to illustrate the proper method for configuring a LISP multihomed site.

Example configuration for HQ-RTR-1 with an xTR, a map server and a map resolver:

```
hostname HQ-RTR-1
!
vrf definition TRANS
 address-family ipv4
 exit
!
vrf definition SOC
address-family ipv4
 exit
!
vrf definition FIN
address-family ipv4
 exit
!
interface Loopback0
 description Management Loopback (in default space)
 ip address 172.31.1.11 255.255.255.255
!
interface GigabitEthernet0/0/0
 description WAN Link to IPv4 Core
 ip address 172.16.1.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 vrf forwarding TRANS
 ip address 10.1.1.1 255.255.255.0
 negotiation auto
```

```
!
interface GigabitEthernet0/0/2
 vrf forwarding SOC
 ip address 10.2.1.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/3
 vrf forwarding FIN
 ip address 10.3.1.1 255.255.255.0
 negotiation auto
!
router lisp
 eid-table default instance-id 0
  database-mapping 172.31.1.11/32 172.16.1.2 priority 1 weight 50
  database-mapping 172.31.1.11/32 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key DEFAULT-key
  ipv4 etr map-server 172.16.1.6 key DEFAULT-key
  exit
 !
 eid-table vrf TRANS instance-id 1
  database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key TRANS-key
  ipv4 etr map-server 172.16.1.6 key TRANS-key
  exit
 !
 eid-table vrf SOC instance-id 2
  database-mapping 10.2.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.2.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key SOC-key
  ipv4 etr map-server 172.16.1.6 key SOC-key
  exit
 !
 eid-table vrf FIN instance-id 3
  database-mapping 10.3.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.3.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key FIN-key
  ipv4 etr map-server 172.16.1.6 key FIN-key
  exit
 !
 site DEFAULT
  authentication-key DEFAULT-key
  eid-prefix 172.31.1.0/24 accept-more-specifics
  exit
 !
 site TRANS
  authentication-key TRANS-key
  eid-prefix instance-id 1 10.1.0.0/16 accept-more-specifics
  exit
 !
 site SOC
  authentication-key SOC-key
  eid-prefix instance-id 2 10.2.0.0/16 accept-more-specifics
  exit
 !
 site FIN
  authentication-key FIN-key
  eid-prefix instance-id 3 10.3.0.0/16 accept-more-specifics
  exit
 !
 ipv4 map-server
 ipv4 map-resolver
 ipv4 itr map-resolver 172.16.1.2
 ipv4 itr map-resolver 172.16.1.6
```

```
 ipv4 itr
 ipv4 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
```

Example configuration for HQ-RTR-2 with an xTR, a map server and a map resolver:

```
hostname HQ-RTR-2
!
vrf definition TRANS
address-family ipv4
 exit
!
vrf definition SOC
address-family ipv4
 exit
!
vrf definition FIN
address-family ipv4
 exit
!
interface Loopback0
 description Management Loopback (in default space)
 ip address 172.31.1.12 255.255.255.255
!
interface GigabitEthernet0/0/0
 description WAN Link to IPv4 Core
 ip address 172.16.1.6 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 vrf forwarding TRANS
 ip address 10.1.1.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/2
 vrf forwarding SOC
 ip address 10.2.1.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/3
 vrf forwarding FIN
 ip address 10.3.1.2 255.255.255.0
 negotiation auto
!
router lisp
 eid-table default instance-id 0
  database-mapping 172.31.1.12/32 172.16.1.2 priority 1 weight 50
  database-mapping 172.31.1.12/32 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key DEFAULT-key
  ipv4 etr map-server 172.16.1.6 key DEFAULT-key
  exit
 !
 eid-table vrf TRANS instance-id 1
  database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key TRANS-key
  ipv4 etr map-server 172.16.1.6 key TRANS-key
  exit
 !
 eid-table vrf SOC instance-id 2
  database-mapping 10.2.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.2.1.0/24 172.16.1.6 priority 1 weight 50
```

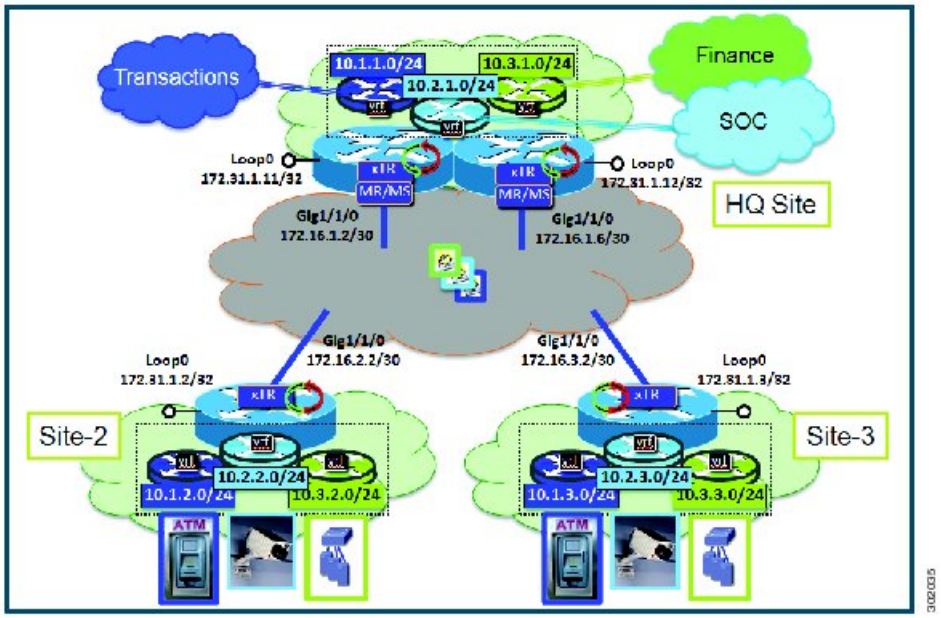
```
 ipv4 etr map-server 172.16.1.2 key SOC-key
 ipv4 etr map-server 172.16.1.6 key SOC-key
 exit
!
eid-table vrf FIN instance-id 3
 database-mapping 10.3.1.0/24 172.16.1.2 priority 1 weight 50
 database-mapping 10.3.1.0/24 172.16.1.6 priority 1 weight 50
 ipv4 etr map-server 172.16.1.2 key FIN-key
 ipv4 etr map-server 172.16.1.6 key FIN-key
 exit
!
site DEFAULT
 authentication-key DEFAULT-key
 eid-prefix 172.31.1.0/24 accept-more-specifics
 exit
!
site TRANS
 authentication-key TRANS-key
 eid-prefix instance-id 1 10.1.0.0/16 accept-more-specifics
 exit
!
site SOC
 authentication-key SOC-key
 eid-prefix instance-id 2 10.2.0.0/16 accept-more-specifics
 exit
!
site FIN
 authentication-key FIN-key
 eid-prefix instance-id 3 10.3.0.0/16 accept-more-specifics
 exit
!
ipv4 map-server
ipv4 map-resolver
ipv4 itr map-resolver 172.16.1.2
ipv4 itr map-resolver 172.16.1.6
ipv4 itr
ipv4 etr
exit
!
ip route 0.0.0.0 0.0.0.0 172.16.1.5
```

Configure a Remote Site for Large-Scale LISP Shared Model Virtualization

Perform this task to enable and configure LISP ITR/ETR (xTR) functionality at a remote site to implement LISP shared model virtualization as part of a large-scale, multiple-site LISP topology.

The configuration demonstrated in this task is part of a more complex, larger scale LISP virtualization solution. In this task, the configuration applies to one of the remote sites shown in the figure below. In this task, the remote site routers only act as xTRs, and use the MS/MRs at the HQ site for LISP control plane support.

Figure 177: Large Scale LISP Site with Virtualized IPv4 EIDs and a Shared IPv4 Core



The components illustrated in the topology shown in the figure above are described below:

- **LISP remote sites:**
  - Each remote site CPE router functions as a LISP ITR and ETR (xTR).
  - Each LISP xTRs has the same three VRFs as the HQ Site: TRANS, SOC, and FIN. Each VRF contains only IPv4 EID-prefixes.
  - Each remote site LISP xTR has a single RLOC connection to a shared IPv4 core network.

Before you begin

The configuration below assumes that the referenced VRFs were created using the **vrf definition** command and that the Configure a Large-Scale LISP Shared Model Virtualization task has been performed at one or more central (headquarters) sites.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **eid-table vrf***vrf-name* **instance-id** *instance-id*
4. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
5. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
6. Repeat Step 13 to configure another locator address for the same LISP map server
7. **exit**
8. **ipv4 itr map-resolver** *map-resolver-address*
9. Repeat Step 16 to configure another locator address for the LISP map resolver
10. **ipv4 itr**

11. **ipv4 etr**
12. **exit**
13. **ip route** *ipv4-prefix next-hop*
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS XE software only).
Step 3	**eid-table vrf***vrf-name* **instance-id** *instance-id* **Example:** `Router(config-router-lisp)# eid-table vrf TRANS instance-id 1`	Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode. • In this example, the VRF table TRANS and instance-id 1 are associated together.
Step 4	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp-eid-table)# database-mapping 10.1.2.0/24 172.16.2.2 priority 1 weight 100`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this example, the EID prefix 10.1.2.0/24 within instance-id 1 at this site is associated with the local IPv4 RLOC 172.16.2.2. **Note** Repeat this step until all EID-to-RLOC mappings within this eid-table vrf and instance ID for the LISP site are configured.
Step 5	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.2 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, the map server and authentication-key are specified here, within the eid-table subcommand mode, so that the authentication key is associated only with this instance ID, within this VPN. **Note** The map server must be configured with EID prefixes and instance-ids matching the one(s) configured on this ETR, as well as an identical authentication key.

	Command or Action	Purpose
		Note The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 6	Repeat Step 13 to configure another locator address for the same LISP map server **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.6 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, a redundant map server is configured. (Because the MS is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for registration (and its neighbor xTR/MS/MR at the same site).
Step 7	**exit** **Example:** `Router(config-router-lisp-eid-table)# exit`	Exits eid-table configuration submode and returns to LISP configuration mode.
Step 8	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.2`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • In this example, the map resolver is specified within router lisp configuration mode and inherited into all eid-table instances since nothing is related to any single instance ID. In addition, redundant map resolvers are configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site). • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)

	Command or Action	Purpose
Step 9	Repeat Step 16 to configure another locator address for the LISP map resolver **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.6`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • In this example, a redundant map resolver is configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site). • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 10	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 11	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 12	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 13	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:

	Command or Action	Purpose
		• a current map-cache entry • a default route with a legitimate next-hop • no route at all In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.
Step 14	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

The example below show the complete configuration for the remote site device illustrated in the figure above and in this task. Note that only one remote site configuration is shown here.

Example configuration for Site 2 with an xTR, and using the map server and a map resolver from the HQ site:

```
hostname Site2-xTR
!
vrf definition TRANS
address-family ipv4
 exit
!
vrf definition SOC
address-family ipv4
 exit
!
vrf definition FIN
address-family ipv4
 exit
!
interface Loopback0
 description Management Loopback (in default space)
 ip address 172.31.1.2 255.255.255.255
!
interface GigabitEthernet0/0/0
 description WAN Link to IPv4 Core
 ip address 172.16.2.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 vrf forwarding TRANS
 ip address 10.1.2.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/2
 vrf forwarding SOC
 ip address 10.2.2.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/3
```

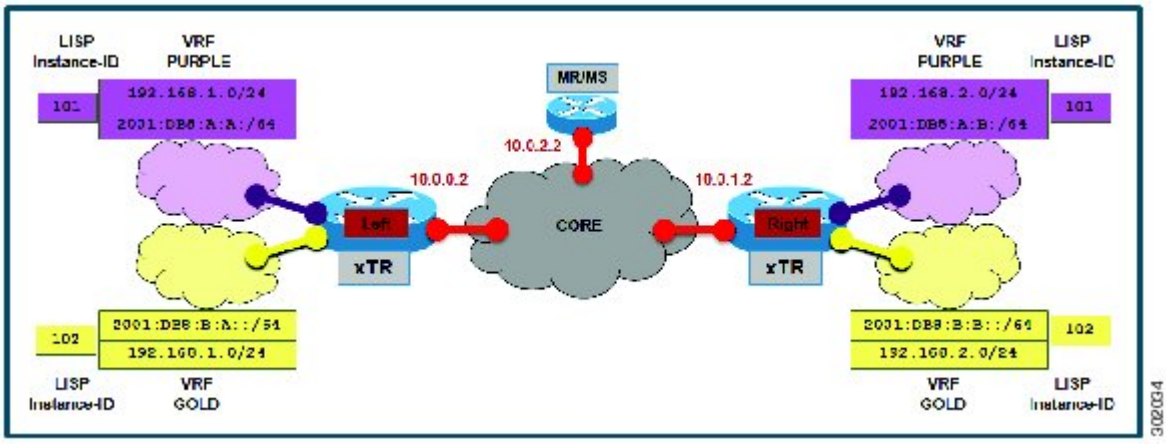
```
 vrf forwarding FIN
 ip address 10.3.2.1 255.255.255.0
 negotiation auto
!
router lisp
 eid-table default instance-id 0
  database-mapping 172.31.1.2/32 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key DEFAULT-key
  ipv4 etr map-server 172.16.1.6 key DEFAULT-key
  exit
 !
 eid-table vrf TRANS instance-id 1
  database-mapping 10.1.2.0/24 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key TRANS-key
  ipv4 etr map-server 172.16.1.6 key TRANS-key
  exit
 !
 eid-table vrf SOC instance-id 2
  database-mapping 10.2.2.0/24 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key SOC-key
  ipv4 etr map-server 172.16.1.6 key SOC-key
  exit
 !
 eid-table vrf FIN instance-id 3
  database-mapping 10.3.2.0/24 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key FIN-key
  ipv4 etr map-server 172.16.1.6 key FIN-key
  exit
 !
 ipv4 itr map-resolver 172.16.1.2
 ipv4 itr map-resolver 172.16.1.6
 ipv4 itr
 ipv4 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 172.16.2.1
```

Verifying and Troubleshooting LISP Virtualization

After configuring LISP, verifying and troubleshooting LISP configuration and operations may be performed by following the optional steps described below. Note that certain verification and troubleshooting steps may only apply to certain types of LISP devices.

In this task, the topology is shown in the figure below and the configuration is from the “Configure Simple LISP Shared Model Virtualization” task, but the commands are applicable to both LISP shared and parallel model virtualization.

Figure 178: Simple LISP Site with Virtualized IPv4 and IPv6 EIDs and a Shared IPv4 Core



Note The following examples do not show every available command and every available output display. Refer to the *Cisco IOS LISP Command Reference* for detailed explanations of each command.

SUMMARY STEPS

1. **enable**
2. **show running-config | section router lisp**
3. **show** [**ip** | **ipv6**] **lisp**
4. **show** [**ip** | **ipv6**] **lisp map-cache**
5. **show** [**ip** | **ipv6**] **lisp database** [**eid-table vrf** *vrf-name*]
6. **show lisp site** [**name** *site-name*]
7. **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}
8. **ping** {*hostname* | *destination-EID*}
9. **clear** [**ip** | **ipv6**] **lisp map-cache**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 show running-config | section router lisp

The **show running-config | section router lisp** command is useful for quickly verifying the LISP configuration on the device. This command applies to any Cisco IOS XE LISP device. The following is sample output from the **show running-config | section router lisp** command when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and a shared IPv4 core:

Example:

```
Router# show running-config | section router lisp

router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:A:A::/64 10.0.0.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:B:A::/64 10.0.0.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Left-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Left-key
 ipv6 etr
 exit
```

Step 3 **show** [**ip** | **ipv6**] **lisp**

The **show ip lisp** and **show ipv6 lisp** commands are useful for quickly verifying the operational status of LISP as configured on the device, as applicable to the IPv4 and IPv6 address families respectively. This command applies to any IOS XE LISP device.

Example:

The first example shows a summary of LISP operational status and IPv6 address family information by EID table:

```
Router# show ipv6 lisp eid-table summary

Instance count: 2
Key: DB - Local EID Database entry count (@ - RLOC check pending
                                          * - RLOC consistency problem),
     DB no route - Local EID DB entries with no matching RIB route,
     Cache - Remote EID mapping cache size, IID - Instance ID,
     Role - Configured Role

                      Interface    DB  DB no  Cache Incom Cache
EID VRF name             (.IID)  size  route   size plete  Idle Role
PURPLE                LISP0.101     1      0      1  0.0%  0.0% ITR-ETR
GOLD                  LISP0.102     1      0      1  0.0%  0.0% ITR-ETR
```

Example:

The second example shows LISP operational status and IPv6 address family information for the VRF named PURPLE:

```
Router# show ipv6 lisp eid-table vrf PURPLE

  Instance ID:                    101
  Router-lisp ID:                 0
  Locator table:                  default
  EID table:                      PURPLE
  Ingress Tunnel Router (ITR):    enabled
  Egress Tunnel Router (ETR):     enabled
  Proxy-ITR Router (PITR):        disabled
  Proxy-ETR Router (PETR):        disabled
  Map Server (MS):                disabled
  Map Resolver (MR):              disabled
```

```
  Map-Request source:               2001:DB8:A:A::1
  ITR Map-Resolver(s):              10.0.2.2
  ETR Map-Server(s):                10.0.2.2 (00:00:24)
  ITR use proxy ETR RLOC(s):        none
```

Example:

The third example shows LISP operational status and IPv6 address family information for the instance ID of 101:

```
Router# show ipv6 lisp instance-id 101

  Instance ID:                      101
  Ingress Tunnel Router (ITR):      enabled
  Egress Tunnel Router (ETR):       enabled
  Proxy-ITR Router (PITR):          disabled
  Proxy-ETR Router (PETR):          disabled
  Map Server (MS):                  disabled
  Map Resolver (MR):                disabled
  Map-Request source:               2001:DB8:A:A::1
  ITR Map-Resolver(s):              10.0.2.2
  ETR Map-Server(s):                10.0.2.2 (00:00:11)
  ITR Solicit Map Request (SMR):    accept and process
    Max SMRs per map-cache entry:   8 more specifics
    Multiple SMR suppression time:  60 secs
  ETR accept mapping data:          disabled, verify disabled
  ETR map-cache TTL:                1d00h
```

Step 4 **show** [**ip** | **ipv6**] **lisp map-cache**

The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands are useful for quickly verifying the operational status of the map cache on a device configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families respectively.

Example:

The following example shows IPv6 mapping cache information based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and a shared IPv4 core. This example output assumes that a map-cache entry has been received for another site with the IPv6 EID prefix 2001:db8:b:b::/64.

```
Router# show ip lisp map-cache eid-table vrf GOLD

LISP IPv6 Mapping Cache for EID-table vrf GOLD (IID 102), 2 entries

::/0, uptime: 01:09:52, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2001:DB8:B:B::/64, uptime: 00:00:10, expires: 23:59:42, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:00:10  up           1/1
```

Step 5 **show** [**ip** | **ipv6**] **lisp database** [**eid-table vrf** *vrf-name*]

The **show ip lisp database** and **show ipv6 lisp database** commands are useful for quickly verifying the operational status of the database mapping on a device configured as an ETR, as applicable to the IPv4 and IPv6 address families respectively.

Example:

The following example shows IPv6 mapping database information for the VRF named GOLD.

```
Router# show ipv6 lisp database eid-table vrf GOLD
```

```
LISP ETR IPv6 Mapping Database for EID-table vrf GOLD (IID 102), LSBs: 0x1, 1 entries

EID-prefix: 2001:DB8:B:A::/64
  10.0.0.2, priority: 1, weight: 1, state: site-self, reachable
```

Step 6 **show lisp site** [**name** *site-name*]

The **show lisp site** command is useful for quickly verifying the operational status of LISP sites, as configured on a map server. This command only applies to a device configured as a map server. The following example output is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and shows the information for the instance ID of 101.

Example:

```
Router# show lisp site instance-id 101

LISP Site Registration Information

Site Name      Last      Up   Who Last            Inst     EID Prefix
               Register       Registered          ID
Left           00:00:36  yes  10.0.0.2            101      192.168.1.0/24
               00:00:43  yes  10.0.0.2            101      2001:DB8:A:A::/64
Right          00:00:31  yes  10.0.1.2            101      192.168.2.0/24
               00:00:02  yes  10.0.1.2            101      2001:DB8:A:B::/64
```

Example:

This second example shows LISP site information for the IPv6 EID prefix of 2001:db8:a:a:/64 and instance ID of 101.

```
Router# show lisp site 2001:db8:a:a:/64 instance-id 101

LISP Site Registration Information

Site name: Left
Allowed configured locators: any
Requested EID-prefix:
  EID-prefix: 2001:DB8:A:A::/64 instance-id 101
    First registered:     02:41:55
    Routing table tag:    0
    Origin:               Configuration
    Registration errors:
      Authentication failures:   4
      Allowed locators mismatch: 0
    ETR 10.0.0.2, last registered 00:00:22, no proxy-reply, no map-notify
                  TTL 1d00h
      Locator   Local  State      Pri/Wgt
      10.0.0.2  yes    up           1/1
```

Step 7 **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}

The LISP Internet Groper (lig) command is useful for testing the LISP control plane. The **lig** command can be used to query for the indicated destination hostname or EID, or the routers local EID-prefix. This command provides a simple means of testing whether a destination EID exists in the LISP mapping database system, or your site is registered with the mapping database system. This command is applicable for both the IPv4 and IPv6 address families and applies to any IOS XE LISP device that maintains a map cache (for example, if configured as an ITR or PITR). The following example output is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and shows the information for the instance ID of 101 and the IPv4 EID prefix of 192.168.2.1.

Example:

```
Router# lig instance-id 101 192.168.2.1
```

```
Mapping information for EID 192.168.2.1 from 10.0.1.2 with RTT 12 msecs
192.168.2.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:00:00  up           1/1
```

Example:

This second example output shows information about the VRF named PURPLE:

```
Router# lig eid-table vrf PURPLE self

Mapping information for EID 192.168.1.0 from 10.0.0.1 with RTT 20 msecs
192.168.1.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.0.0.1  00:00:00  up, self     1/1
```

Step 8 **ping** {*hostname* | *destination-EID*}

The **ping** command is useful for testing basic network connectivity and reachability and/or liveness of a destination EID or RLOC address. When using **ping** it is important to be aware that because LISP uses an encapsulation, you should always specify a source address; never allow the **ping** application to assign its own default source address. This is because there are four possible ways to use **ping**, and without explicitly indicating the source address, the wrong one may be used by the application leading to erroneous results that complicate operational verification or troubleshooting. The four possible uses of **ping** include:

- RLOC-to-RLOC—Sends “echo?? packets out natively (no LISP encap) and receive the “echo-reply?? back natively. This can be used to test the underlying network connectivity between locators of various devices, such as xTR to Map-Server or Map-Resolver.
- EID-to-EID—Sends “echo?? packets out LISP-encaped and receive the “echo-reply?? back LISP-encaped. This can be used to test the LISP data plane (encapsulation) between LISP sites.
- EID-to-RLOC—Sends “echo?? packets out natively (no LISP encap) and receive the "echo-reply" back LISP-encaped through a PITR mechanism. This can be used to test the PITR infrastructure.
- RLOC-to-EID - Sends “echo?? packets out LISP-encaped and receive the “echo-reply?? back natively. This can be used to test PETR capabilities.

The **ping** command is applicable to the IPv4 and IPv6 address families respectively, and can be used on any IOS XE LISP device in some manner. (The ability to do LISP encapsulation, for example, requires the device to be configured as an ITR or PITR.)

The following example output from the **ping** command is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes. (Note that ping is not a LISP command and does not know about an EID table or an instance ID. When virtualization is included, output limiters can only be specified by VRF.)

Example:

```
Router# ping vrf PURPLE 2001:DB8:a:b::1 source 2001:DB8:a:a::1 rep 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 2001:DB8:A:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:A:A::1%PURPLE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 0/0/1 ms
```

Example:

```
Router# ping vrf GOLD

Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:b:b::1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: y
Source address or interface: 2001:db8:b:a::1
.
.
.
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:B:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:B:A::1%GOLD
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Step 9 **clear** [**ip** | **ipv6**] **lisp map-cache**

The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries stored by the router. This can be useful trying to quickly verify the operational status of the LISP control plane. This command applies to a LISP device that maintains a map cache (for example, if configured as an ITR or PITR).

Example:

The following example displays IPv4 mapping cache information for instance ID 101, shows the command used to clear the mapping cache for instance ID 101, and displays the show information after clearing the cache.

```
Router# show ip lisp map-cache instance-id 101

LISP IPv4 Mapping Cache for EID-table vrf PURPLE (IID 101), 2 entries

0.0.0.0/0, uptime: 00:25:17, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
192.168.2.0/24, uptime: 00:20:13, expires: 23:39:39, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:20:13  up           1/1

Router# clear ip lisp map-cache instance-id 101

Router# show ip lisp map-cache instance-id 101

LISP IPv4 Mapping Cache, 1 entries

0.0.0.0/0, uptime: 00:00:02, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

Configuration Examples for LISP Shared Model Virtualization

Complete configuration examples are available within each task under the “How to Configure LISP Shared Model Virtualization” section.

Additional References

Related Documents

Document Title	Location
Cisco IOS IP Routing: LISP Command Reference	http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_lisp/command/ip-lisp-cr-book.html
Enterprise IPv6 Transitions Strategy Using the Locator/ID Separation Protocol	Cisco LISP Software Image Download Page
Cisco IOS LISP0 Virtual Interface, Application Note, Version 1.0	Cisco LISP Software Image Download Page
Cross-Platform Release Notes for Cisco IOS Release 15.2M&T	http://www.cisco.com/en/US/docs/ios/15_2m_and_t/release/notes/15_2m_and_t.html

Standards

Standard	Title
IANA Address Family Numbers	http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xml

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
draft-ietf-lisp-22	Locator/ID Separation Protocol (LISP) http://tools.ietf.org/html/draft-ietf-lisp-22
draft-ietf-lisp-ms-16	LISP Map Server http://tools.ietf.org/html/draft-ietf-lisp-ms-16
draft-ietf-lisp-alt-10	LISP Alternative Topology (LISP+ALT) http://tools.ietf.org/html/draft-ietf-lisp-alt-10
draft-ietf-lisp-LCAF-06	LISP Canonical Address Format (LCAF) http://tools.ietf.org/wg/lisp/
draft-ietf-lisp-interworking-06	Interworking LISP with IPv4 and IPv6 http://tools.ietf.org/html/draft-ietf-lisp-interworking-06
draft-ietf-lisp-lig-06	LISP Internet Groper (LIG) http://tools.ietf.org/html/draft-ietf-lisp-lig-06
draft-ietf-lisp-mib-03	LISP MIB http://tools.ietf.org/wg/lisp/draft-ietf-lisp-mib/

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Shared Model Virtualization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 194: Feature Information for LISP Shared Model Virtualization

Feature Name	Releases	Feature Information
LISP Shared Model Virtualization	15.2(2)T 15.1(1)SY1	LISP Shared Model Virtualization feature uses Endpoint Identifier (EID) spaces that are created by binding VRFs associated with an EID space to Instance IDs. A common, "shared" locator space is used by all virtualized EIDs.

CHAPTER 181

LISP Parallel Model Virtualization

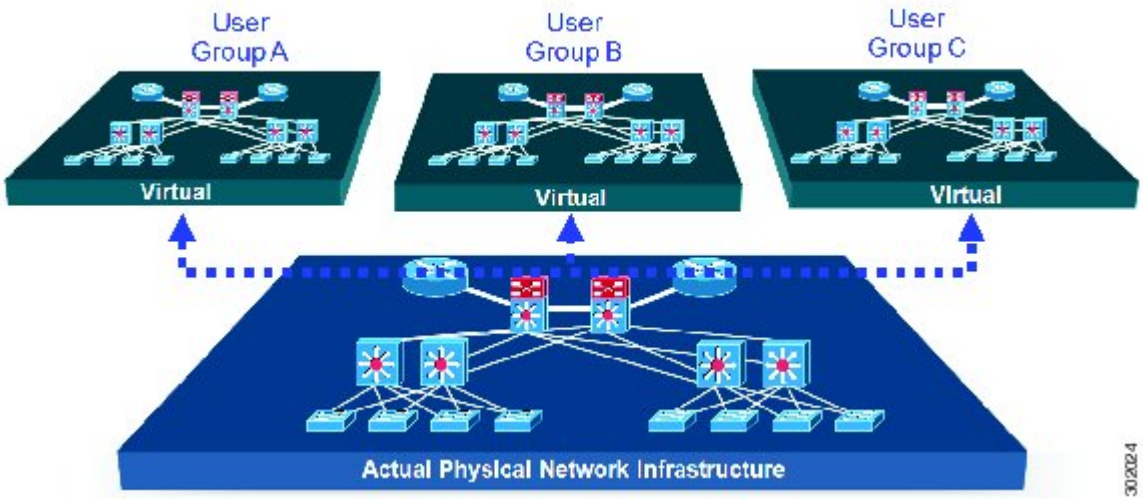
Information About LISP Parallel Model Virtualization

Overview of LISP Virtualization

Deploying physical network infrastructure requires both capital investments for hardware, as well as manpower investments for installation and operational management support. When distinct user groups within an organization desire to control their own networks, it rarely makes economic sense for these user groups to deploy and manage separate physical networks. Physical plants are rarely utilized to their fullest, resulting in stranded capacity (bandwidth, processor, memory, etc.). In addition, the power, rack space, and cooling needs to physical plants do not satisfy modern “green” requirements. Network virtualization offers the opportunity to satisfy organizational needs, while efficiently utilizing physical assets.

The purpose of network virtualization, as shown in the figure below, is to create multiple, logically separated topologies across one common physical infrastructure.

Figure 179: LISP Deployment Environment



When considering the deployment of a virtualized network environment, take into account both the device and the path level.

Device Level Virtualization

Virtualization at the device level entails the use of the virtual routing and forwarding (VRF) to create multiple instances of Layer 3 routing tables, as illustrated in the figure below. VRFs provide segmentation across IP addresses, allowing for overlapped address space and traffic separation. Separate routing, QoS, security, and management policies can be applied to each VRF instance. An IGP or EGP routing process is typically enabled within a VFR, just as it would be in the global (default) routing table. As described in detail below, LISP binds VRFs to instance IDs for similar purposes.

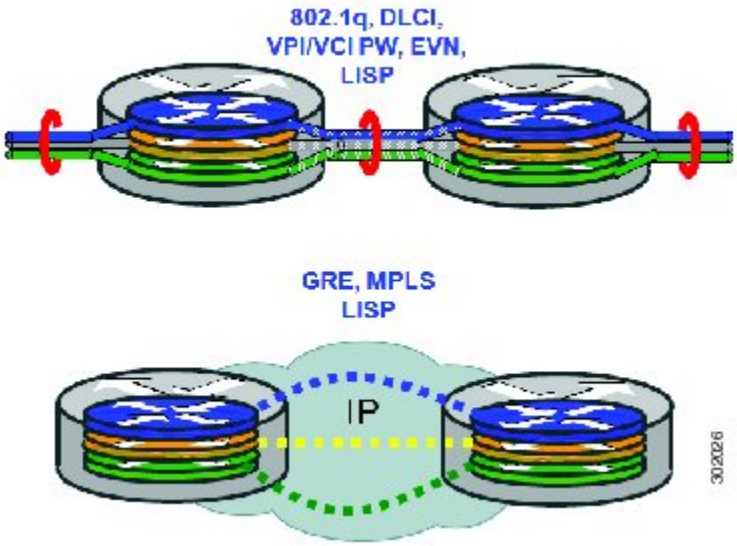
Figure 180: Device Level Virtualization

Path Level Virtualization

VRF table separation is maintained across network paths using any number of traditional mechanisms, as illustrated in the figure below. Single-hop path segmentation (hop-by-hop) is typically accomplished by techniques such as 802.1q VLANs, VPI/VCI PW, or EVN. LISP can also be used. Traditional multi-hop

mechanisms include MPLS and GRE tunnels. As described in detail below, LISP binds VRFs to instance IDs (IIDs), and then these IIDs are included in the LISP header to provide data plane (traffic flow) separation for single or multihop needs.

Figure 181: Path Level Virtualization



LISP Virtualization at the Device Level

Recalling that LISP implements Locator ID separation and, in so doing, creates two namespaces (EIDs and RLOCs), it is easy to see that LISP virtualization can consider both EID and RLOC namespaces for virtualization. That is, either or both can be virtualized.

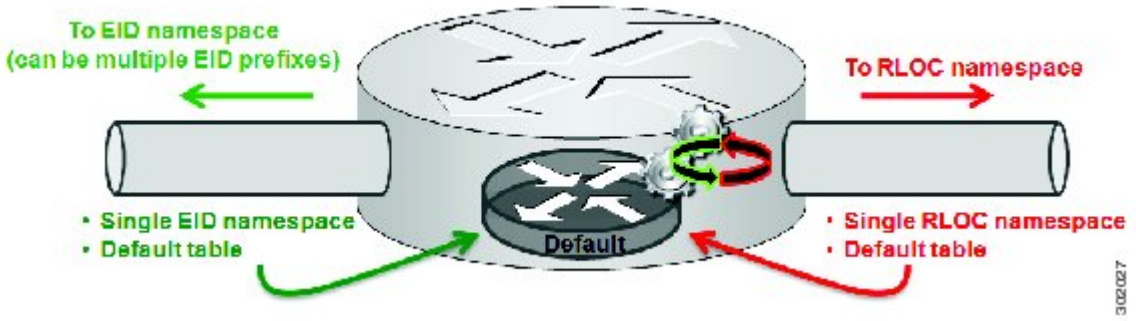
- EID virtualization—Enabled by binding a LISP instance ID to an EID VRF. Instance IDs are numerical tags defined in the LISP canonical address format (LCAF) draft, and are used to maintain address space segmentation in both the control plane and data plane.
- RLOC virtualization—Tying locator addresses and associated mapping services to the specific VRF within which they are reachable enables RLOC virtualization.

Because LISP considers virtualization of both EID and RLOC namespaces, two models of operation are defined: shared model and parallel model. For completeness, the discussions below begin first with a review of the default (non-virtualized) model of LISP, and then cover the details of shared and parallel models.

Default (Non-Virtualized) LISP Model

By default, LISP is not virtualized in either EID space or RLOC space. That is, unless otherwise configured, both EID and RLOC addresses are resolved in the default (global) routing table. This concept is illustrated in the figure below.

Figure 182: Default (Non-Virtualized) LISP Model (Resolves Both EID and RLOC Addresses in the Default (Global) Routing Table.

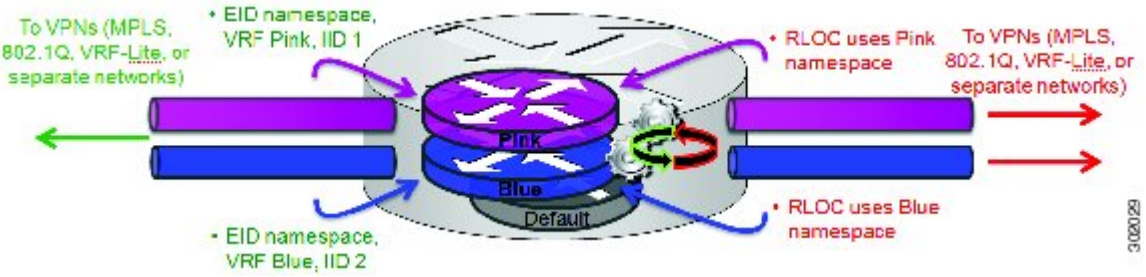


As shown in the figure above, both EID and RLOC addresses are resolved in the default table. The mapping system must also be reachable via the default table. This default model can be thought of as a single instantiation of the parallel model of LISP virtualization where EID and RLOC addresses are within the same namespace such as is the case in this default table.

LISP Parallel Model Virtualization

LISP parallel model virtualization ties virtualized EID space associated with VRFs to RLOCs associated with the same or different VRFs. This concept is illustrated in the figure below.

Figure 183: LISP parallel model virtualization resolves an EID and associated RLOCs within the same or different VRF. In this example, both EID and RLOC addresses are resolved in the same VRF, but multiple (parallel) segmentation is configured on the same device (BLUE and PINK).

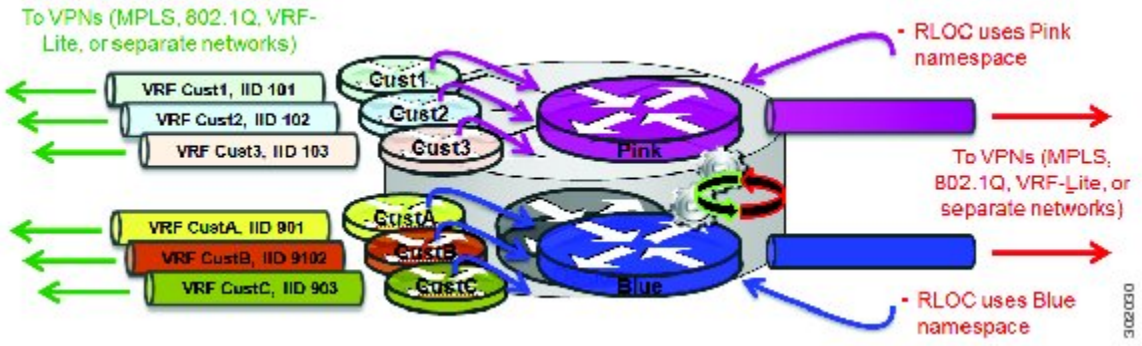


As shown in the figure above, EID space is virtualized through its association with VRFs, and these VRFs are tied to LISP Instance IDs to segment the control plane and data plane in LISP. A common, “shared” locator space, the default (global) table as shown in the figure above, is used to resolve RLOC addresses for all virtualized EIDs. The mapping system must also be reachable via the common locator space as well.

The example illustrated in the figure above shows virtualized EID space associated with a VRF (and bound to an Instance ID) being tied to locator space associated with the same VRF, in this case - Pink/Pink and Blue/Blue. However, this is not required; the EID VRF does not need to match the RLOC VRF. In any case, a mapping system must be reachable via the associated locator space. Multiple parallel instantiations can be defined.

In the most general case, shared model and parallel model may be combined such that multiple EID VRFs share a common RLOC VRF, and multiple instantiations of this architecture are implemented on the same platform, as shown in the figure below.

Figure 184: LISP shared and parallel models may be combined for maximum flexibility.

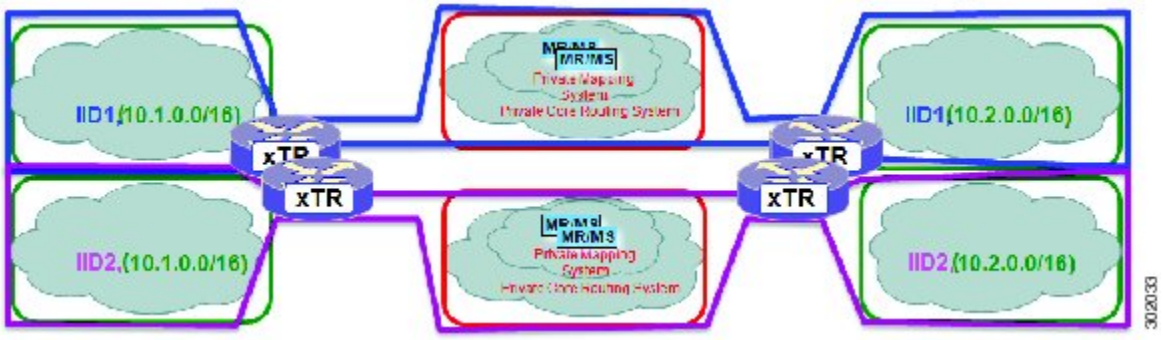


As shown in the figure above, shared and parallel models are combined to associate several EID instances to one shared RLOC VRF, and then several other EID instances to another shared RLOC VRF.

LISP Parallel Model Virtualization Architecture

Architecturally, LISP parallel model virtualization can be deployed in single or multitenancy configurations. In the parallel model multitenancy case, a set of xTRs is shared (virtualized) among multiple customers, and each customer uses their own private (segmented) core infrastructure and mapping system. All sites associated with the customer use the same instance ID and are part of a VPN using their own EID namespace as shown in the figure below.

Figure 185: In the LISP parallel model multitenancy case, shared xTRs use virtualized core networks and mapping systems. LISP instance IDs segment the LISP data plane and control plane.



LISP Parallel Model Virtualization Implementation Considerations and Caveats

When the LISP Parallel Model Virtualization is implemented, several important considerations and caveats are important. Each **router lisp** *value* instantiation is considered by Cisco IOS XE software to be a separate process. Instance IDs must be unique only within a **router lisp** instantiation. Review the example below:

```
xTR-1(config)# vrf definition alpha
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config)# vrf definition beta
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config-vrf)# vrf definition gamma
```

```
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config-vrf)# vrf definition delta
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config-vrf)# exit
xTR-1(config)# router lisp 1
xTR-1(config-router-lisp)# locator-table vrf alpha
xTR-1(config-router-lisp)# eid-table vrf beta instance-id 101
xTR-1(config-router-lisp-eid-table)# exit
xTR-1(config-router-lisp)# exit
xTR-1(config)# router lisp 2
xTR-1(config-router-lisp)# locator-table vrf gamma
xTR-1(config-router-lisp)# eid-table vrf delta instance-id 101
xTR-1(config-router-lisp-eid-table)# exit
xTR-1(config-router-lisp)# eid-table vrf beta instance-id 201
The vrf beta table is not available for use as an EID table (in use by router lisp 1 EID
instance 101 VRF)
```

In the above example, four VRFs are created; alpha, beta, gamma, and delta. The **router lisp** instantiation router lisp 1 is created and associated with the locator-table VRF named alpha. Next, the EID table VRF named beta is specified and associated with instance ID 101. Next, a new **router lisp** instantiation, router lisp 2, is created and associated with the locator-table VRF named gamma. Next, EID table VRF named delta is specified and also associated with instance ID 101. These two instance IDs are unrelated to each other; one is relevant only within router lisp 1 and the other is only relevant within router lisp 2.

In the above example, also observe that while under router lisp 2, an attempt is made to configure an EID table VRF named beta. Note that the router is unable to use this EID table VRF since it (beta) is already associated with an **eid-table** command within the router lisp 1 instantiation.

You can re-use an instance ID, and which EID VRF it is decapsulated into depends on the **router lisp** instantiation and locator-table VRF that it is associated with. You cannot connect the same EID VRF to more than one locator-table VRF, however.

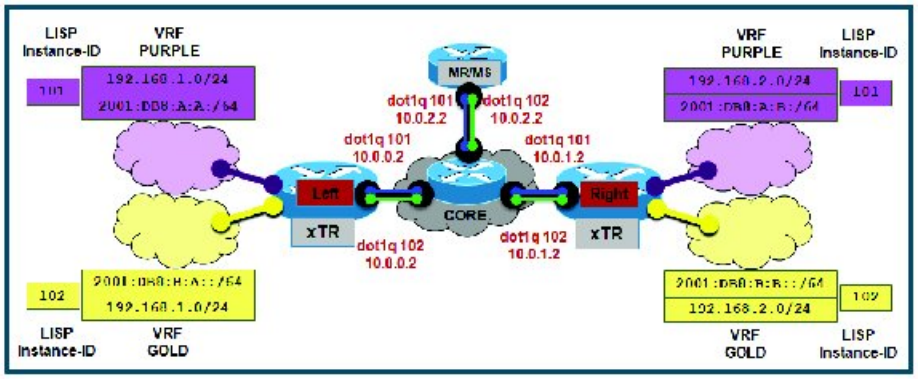
How to Configure LISP Parallel Model Virtualization

Configure Simple LISP Parallel Model Virtualization

Perform these tasks to enable and configure LISP ITR/ETR (xTR) functionality and LISP map resolver and map server for LISP parallel model virtualization.

The configuration implemented in this task and illustrated in the figure below is for two LISP sites that are connected in parallel mode. Each LISP site uses a single edge router configured as both an ITR and ETR (xTR), with a single connection to its upstream provider. However, the upstream connection is VLAN-segmented to maintain RLOC space separation within the core. Two VRFs are defined here: BLUE and GREEN. IPv4 RLOC space is used in each of these parallel networks. Both IPv4 and IPv6 EID address space is used. The LISP site registers to one map server/map resolver (MS/MR), which is segmented to maintain the parallel model architecture of the core network.

Figure 186: Simple LISP Site with One IPv4 RLOC and One IPv4 EID



The components illustrated in the topology shown in the figure above are described below:

- **LISP site:**
  - The CPE functions as a LISP ITR and ETR (xTR).
  - Both LISP xTRs have two VRFs: GOLD and PURPLE, with each VRF containing both IPv4 and IPv6 EID-prefixes, as shown in the figure above. Note the overlapping prefixes, used for illustration purposes. A LISP instance-id is used to maintain separation between two VRFs. Note that in this example, the share key is configured “per-VPN.  ?
  - Each LISP xTR has a single RLOC connection to a parallel IPv4 core network.

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map-server and map-resolver for mapping services. The example configurations at the end of this task show the full configuration for two xTRs (Left-xTR and Right-xTR).

Before you begin

The configuration below assumes that the referenced VRFs were created using the **vrf definition** command.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp** *lisp-instantiation-number*
3. **locator-table vrf** *rloc-vrf-name*
4. **eid-table vrf***EID-vrf-name* **instance-id** *instance-id*
5. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
6. Repeat Step 4 until all EID-to-RLOC mappings within this eid-table vrf and instance ID for this LISP site are configured.
7. **exit**
8. **ipv4 itr map-resolver** *map-resolver-address*
9. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
10. **ipv4 itr**
11. **ipv4 etr**
12. **ipv6 itr map-resolver** *map-resolver-address*

13. **ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key*
14. **ipv6 itr**
15. **ipv6 etr**
16. **exit**
17. **ip route vrf** *rloc-vrf-name ipv4-prefix next-hop*
18. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** *lisp-instantiation-number* **Example:** `Router(config)# router lisp`	Creates the specified LISP instantiation number and enters LISP configuration mode (Cisco IOS XE software only). All subsequent LISP commands apply to that router LISP instantiation. • In this example, the router LISP instantiation 1 is configured.
Step 3	**locator-table vrf** *rloc-vrf-name* **Example:** `Router(config-router-lisp)# locator-table vrf BLUE`	Configures a router LISP instantiation to use the specified VRF as RLOC space when encapsulating EIDs and sending control plane packets. • In this example, the RLOC VRF named BLUE is configured.
Step 4	**eid-table vrf***EID-vrf-name* **instance-id** *instance-id* **Example:** `Router(config-router-lisp)# eid-table vrf PURPLE instance-id 101`	Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode. • In this example, the VRF table PURPLE and instance-id 101 are associated together.
Step 5	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp-eid-table)# database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this example, a single IPv4 EID prefix, 192.168.1.0/24, within instance ID 1 at this site is associated with the local IPv4 RLOC 10.0.0.2.
Step 6	Repeat Step 4 until all EID-to-RLOC mappings within this eid-table vrf and instance ID for this LISP site are configured. **Example:** `Router(config-router-lisp-eid-table)#`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this example, the IPv6 EID prefix, 2001:db8:a:a::/64, within instance ID 1 at this site is also associated with the local IPv4 RLOC 10.0.0.2.

	Command or Action	Purpose
	`database-mapping 2001:db8:a:a::/64 10.0.0.2 priority 1 weight 1`	
Step 7	**exit** **Example:** `Router(config-router-lisp-eid-table)# exit`	Exits eid-table configuration submode and returns to LISP configuration mode.
Step 8	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • In this example, the map resolver is specified within router lisp configuration mode. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 9	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.0.2.2 key 0 PURPLE-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, the map server and authentication key are specified within router lisp configuration mode. • The map server must be configured with EID prefixes and instance IDs matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 10	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.

	Command or Action	Purpose
Step 11	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 12	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv6 EID-to-RLOC mapping resolutions. • In this example, the map resolver is specified within router lisp configuration mode. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-resolver is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 13	**ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.0.2.2 key 0 PURPLE-key`	Configures a locator address for the LISP map-server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system. • In this example, the map server and authentication key are specified within router lisp configuration mode. • The map-server must be configured with EID prefixes and instance IDs matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map-server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 14	**ipv6 itr** **Example:** `Router(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 15	**ipv6 etr** **Example:**	Enables LISP ETR functionality for the IPv6 address family.

	Command or Action	Purpose
	`Router(config-router-lisp)# ipv6 etr`	
Step 16	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 17	**ip route vrf** *rloc-vrf-name ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route vrf BLUE 0.0.0.0 0.0.0.0 10.0.0.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

Step 18	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

The examples below show the complete configuration for the LISP topology illustrated in the figure above and in this task. On the xTRs, the VRFs and EID prefixes are assumed to be attached to VLANs configured on the devices.

Example configuration for the Left xTR:

```
hostname Left-xTR
!
ipv6 unicast-routing
!
vrf definition PURPLE
 address-family ipv4
 exit
```

```
 address-family ipv6
 exit
!
vrf definition GOLD
 address-family ipv4
 exit
 address-family ipv6
 exit
!
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0
!
interface Ethernet1/0.1
 encapsulation dot1q 101
 vrf forwarding PURPLE
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:A:A::1/64
!
interface Ethernet1/0.2
 encapsulation dot1q 102
 vrf forwarding GOLD
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:B:A::1/64
!
router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:A:A::/64 10.0.0.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:B:A::/64 10.0.0.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Left-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Left-key
 ipv6 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 10.0.0.1
!
```

Example configuration for Right xTR:

```
hostname Right-xTR
!
ipv6 unicast-routing
!
vrf definition PURPLE
 address-family ipv4
 exit
 address-family ipv6
 exit
!
vrf definition GOLD
 address-family ipv4
 exit
 address-family ipv6
 exit
!
```

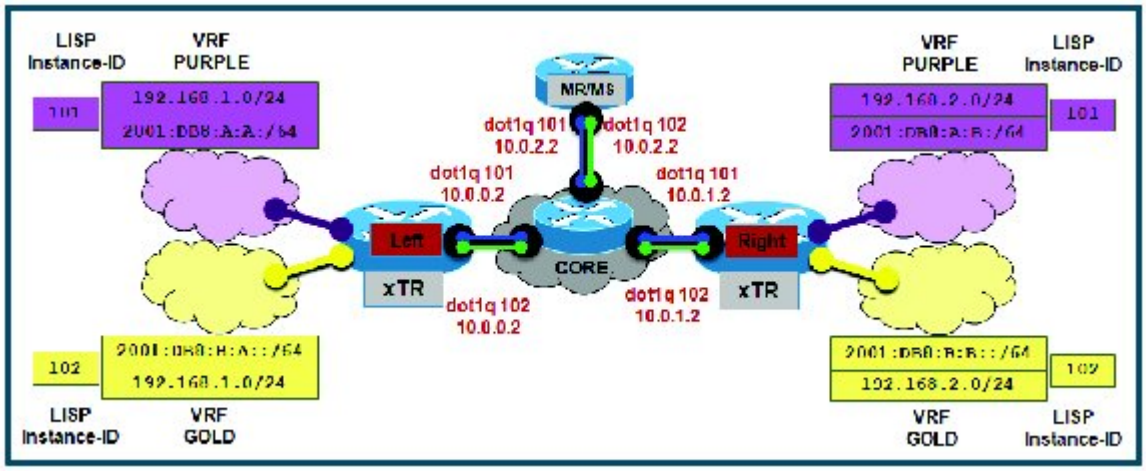
```
interface Ethernet0/0
 ip address 10.0.1.2 255.255.255.0
!
interface Ethernet1/0.1
 encapsulation dot1q 101
 vrf forwarding PURPLE
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:A:B::1/64
!
interface Ethernet1/0.2
 encapsulation dot1q 102
 vrf forwarding GOLD
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:B:B::1/64
!
router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 1
  database-mapping 2001:DB8:A:B::/64 10.0.1.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 1
  database-mapping 2001:DB8:B:B::/64 10.0.1.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Right-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Right-key
 ipv6 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 10.0.1.1
!
```

Configuring a Private LISP Mapping System for LISP Parallel Model Virtualization

Perform this task to configure and enable standalone LISP map server/map resolver functionality for LISP parallel model virtualization. In this task, a Cisco router is configured as a standalone map resolver/map server (MR/MS) for a private LISP mapping system. Because the MR/MS is configured as a stand-alone device, it has no need for LISP alternate logical topology (ALT) connectivity. All relevant LISP sites must be configured to register with this map server so that this map server has full knowledge of all registered EID prefixes within the (assumed) private LISP system.

- **Mapping system:**

Figure 187: Simple LISP Site with One IPv4 RLOC and One IPv4 EID



- One map resolver/map server (MS/MR) system is shown in the figure above and assumed available for the LISP xTR to register to within the proper parallel RLOC space. The MS/MR has an IPv4 RLOC address of 10.0.2.2, within each VLAN/VRF (Green and Blue) providing parallel model RLOX separation in the IPv4 core.
- The map server site configurations are virtualized using LISP instance IDs to maintain separation between the two VRFs, PURPLE and GOLD.

Repeat this task for all router lisp instantiations and RLOC VRFs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp** *lisp-instantiation-number*
4. **locator-table vrf** *rloc-vrf-name*
5. **site** *site-name*
6. **authentication-key** [*key-type*] *authentication-key*
7. **eid-prefix instance-id** *instance-id EID-prefix*
8. **eid-prefix instance-id** *instance-id EID-prefix*
9. **exit**
10. **ipv4 map-resolver**
11. **ipv4 map-server**
12. **ipv6 map-resolver**
13. **ipv6 map-server**
14. **exit**
15. **ip route vrf** *rloc-vrf-name ipv4-prefix next-hop*
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** *lisp-instantiation-number* **Example:** `Router(config)# router lisp`	Creates the specified LISP instantiation number and enters LISP configuration mode (IOS XE software only). All subsequent LISP commands apply to that router LISP instantiation. • In this example, the router LISP instantiation 1 is configured.
Step 4	**locator-table vrf** *rloc-vrf-name* **Example:** `Router(config)# locator-table vrf BLUE`	Configures a router lisp instantiation to use the specified VRF as RLOC space when encapsulating EIDs and sending control plane packets. • In this example, the RLOC VRF BLUE is configured.
Step 5	**site** *site-name* **Example:** `Router(config-router-lisp)# site Purple`	Specifies a LISP site named Purple and enters LISP site configuration mode. • In this example, the LISP site named Purple is configured.
Step 6	**authentication-key** [*key-type*] *authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 Purple-key`	Configures the password used to create the SHA-2 HMAC hash for authenticating the map register messages sent by an ETR when registering to the map server. **Note** The ETR must be configured with EID prefixes and instance IDs matching the one(s) configured on this map server, as well as an identical authentication key.
Step 7	**eid-prefix instance-id** *instance-id EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 101 192.168.1.0/24`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. Repeat this step as necessary to configure additional IPv4 EID prefixes under this LISP site. • In this example, the IPv4 EID prefix 192.168.1.0/24 and instance ID 101 are associated together.

	Command or Action	Purpose
Step 8	**eid-prefix instance-id** *instance-id EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 101 2001:db8:a:a::/64`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. Repeat this step as necessary to configure additional IPv6 EID prefixes under this LISP site. • In this example, the IPv6 EID prefix 2001:db8:a:a::/64 and instance ID 101 are associated together.
Step 9	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 10	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family within this router lisp instantiation.
Step 11	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family within this router lisp instantiation.
Step 12	**ipv6 map-resolver** **Example:** `Router(config-router-lisp)# ipv6 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family within this router lisp instantiation.
Step 13	**ipv6 map-server** **Example:** `Router(config-router-lisp)# ipv6 map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family within this router lisp instantiation.
Step 14	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 15	**ip route vrf** *rloc-vrf-name ipv4-prefix next-hop* **Example:** `Router(config)# ip route vrf BLUE 0.0.0.0 0.0.0.0 10.0.2.1`	Configures a default route to the upstream next hop for all IPv4 destinations, reachable within the specified RLOC VRF.
Step 16	**exit** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Router(config)# exit`	

Example:

Example configuration for the map server/map resolver.

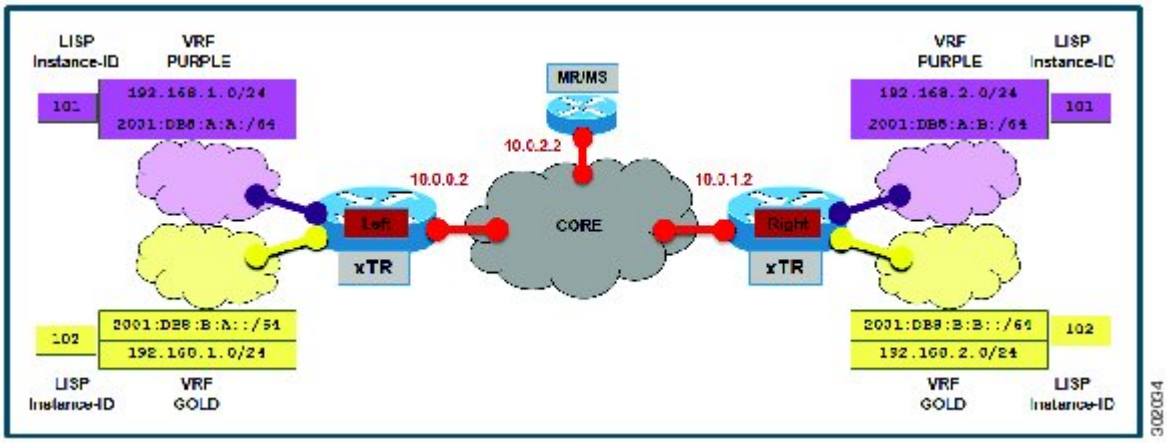
```
hostname MSMR
!
vrf definition BLUE
 address-family ipv4
 exit
!
vrf definition GREEN
 address-family ipv4
 exit
!
ipv6 unicast-routing
!
interface Ethernet0/0.101
 encapsulation dot1Q 101
 vrf forwarding BLUE
 ip address 10.0.0.2 255.255.255.0
!
interface Ethernet0/0.102
 encapsulation dot1Q 102
 vrf forwarding GREEN
 ip address 10.0.0.2 255.255.255.0
!
router lisp 1
 locator-table vrf BLUE
 site Purple
  authentication-key PURPLE-key
  eid-prefix instance-id 101 192.168.1.0/24
  eid-prefix instance-id 101 192.168.2.0/24
  eid-prefix instance-id 101 2001:DB8:A:A::/64
  eid-prefix instance-id 101 2001:DB8:A:B::/64
 !
 ipv4 map-server
 ipv4 map-resolver
 ipv6 map-server
 ipv6 map-resolver
!
router lisp 2
 locator-table vrf GREEN
 site Gold
  authentication-key GOLD-key
  eid-prefix instance-id 102 192.168.1.0/24
  eid-prefix instance-id 102 192.168.2.0/24
  eid-prefix instance-id 102 2001:DB8:B:A::/64
  eid-prefix instance-id 102 2001:DB8:B:B::/64
!
 ipv4 map-server
 ipv4 map-resolver
 ipv6 map-server
 ipv6 map-resolver
!
ip route vrf GREEN 0.0.0.0 0.0.0.0 10.0.2.1
ip route vrf BLUE 0.0.0.0 0.0.0.0 10.0.2.1
```

Verifying and Troubleshooting LISP Virtualization

After configuring LISP, verifying and troubleshooting LISP configuration and operations may be performed by following the optional steps described below. Note that certain verification and troubleshooting steps may only apply to certain types of LISP devices.

In this task, the topology is shown in the figure below and the configuration is from the "Configure Simple LISP Shared Model Virtualization" task, but the commands are applicable to both LISP shared and parallel model virtualization.

Figure 188: Simple LISP Site with Virtualized IPv4 and IPv6 EIDs and a Shared IPv4 Core



Note The following examples do not show every available command and every available output display. Refer to the *Cisco IOS LISP Command Reference* for detailed explanations of each command.

SUMMARY STEPS

1. **enable**
2. **show running-config | section router lisp**
3. **show** [**ip** | **ipv6**] **lisp**
4. **show** [**ip** | **ipv6**] **lisp map-cache**
5. **show** [**ip** | **ipv6**] **lisp database** [**eid-table vrf** *vrf-name*]
6. **show lisp site** [**name** *site-name*]
7. **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}
8. **ping** {*hostname* | *destination-EID*}
9. **clear** [**ip** | **ipv6**] **lisp map-cache**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config | section router lisp**

The **show running-config | section router lisp** command is useful for quickly verifying the LISP configuration on the device. This command applies to any Cisco IOS XE LISP device. The following is sample output from the **show running-config | section router lisp** command when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and a shared IPv4 core:

Example:

```
Router# show running-config | section router lisp

router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:A:A::/64 10.0.0.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:B:A::/64 10.0.0.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Left-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Left-key
 ipv6 etr
 exit
```

Step 3 **show** [**ip** | **ipv6**] **lisp**

The **show ip lisp** and **show ipv6 lisp** commands are useful for quickly verifying the operational status of LISP as configured on the device, as applicable to the IPv4 and IPv6 address families respectively. This command applies to any IOS XE LISP device.

Example:

The first example shows a summary of LISP operational status and IPv6 address family information by EID table:

```
Router# show ipv6 lisp eid-table summary

Instance count: 2
Key: DB - Local EID Database entry count (@ - RLOC check pending
                                          * - RLOC consistency problem),
     DB no route - Local EID DB entries with no matching RIB route,
     Cache - Remote EID mapping cache size, IID - Instance ID,
     Role - Configured Role

                      Interface    DB  DB no  Cache Incom Cache
EID VRF name             (.IID)  size  route   size plete  Idle Role
PURPLE                LISP0.101     1      0      1  0.0%  0.0% ITR-ETR
GOLD                  LISP0.102     1      0      1  0.0%  0.0% ITR-ETR
```

Example:

The second example shows LISP operational status and IPv6 address family information for the VRF named PURPLE:

```
Router# show ipv6 lisp eid-table vrf PURPLE
```

```
  Instance ID:                        101
  Router-lisp ID:                     0
  Locator table:                      default
  EID table:                          PURPLE
  Ingress Tunnel Router (ITR):        enabled
  Egress Tunnel Router (ETR):         enabled
  Proxy-ITR Router (PITR):            disabled
  Proxy-ETR Router (PETR):            disabled
  Map Server (MS):                    disabled
  Map Resolver (MR):                  disabled
  Map-Request source:                 2001:DB8:A:A::1
  ITR Map-Resolver(s):                10.0.2.2
  ETR Map-Server(s):                  10.0.2.2 (00:00:24)
  ITR use proxy ETR RLOC(s):          none
```

Example:

The third example shows LISP operational status and IPv6 address family information for the instance ID of 101:

```
Router# show ipv6 lisp instance-id 101

  Instance ID:                        101
  Ingress Tunnel Router (ITR):        enabled
  Egress Tunnel Router (ETR):         enabled
  Proxy-ITR Router (PITR):            disabled
  Proxy-ETR Router (PETR):            disabled
  Map Server (MS):                    disabled
  Map Resolver (MR):                  disabled
  Map-Request source:                 2001:DB8:A:A::1
  ITR Map-Resolver(s):                10.0.2.2
  ETR Map-Server(s):                  10.0.2.2 (00:00:11)
  ITR Solicit Map Request (SMR):      accept and process
    Max SMRs per map-cache entry:     8 more specifics
    Multiple SMR suppression time:    60 secs
  ETR accept mapping data:            disabled, verify disabled
  ETR map-cache TTL:                  1d00h
```

Step 4 **show** [**ip** | **ipv6**] **lisp map-cache**

The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands are useful for quickly verifying the operational status of the map cache on a device configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families respectively.

Example:

The following example shows IPv6 mapping cache information based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and a shared IPv4 core. This example output assumes that a map-cache entry has been received for another site with the IPv6 EID prefix 2001:db8:b:b::/64.

```
Router# show ip lisp map-cache eid-table vrf GOLD

LISP IPv6 Mapping Cache for EID-table vrf GOLD (IID 102), 2 entries

::/0, uptime: 01:09:52, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2001:DB8:B:B::/64, uptime: 00:00:10, expires: 23:59:42, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:00:10  up           1/1
```

Step 5 **show** [**ip** | **ipv6**] **lisp database** [**eid-table vrf** *vrf-name*]

The **show ip lisp database** and **show ipv6 lisp database** commands are useful for quickly verifying the operational status of the database mapping on a device configured as an ETR, as applicable to the IPv4 and IPv6 address families respectively.

Example:

The following example shows IPv6 mapping database information for the VRF named GOLD.

```
Router# show ipv6 lisp database eid-table vrf GOLD

LISP ETR IPv6 Mapping Database for EID-table vrf GOLD (IID 102), LSBs: 0x1, 1 entries

EID-prefix: 2001:DB8:B:A::/64
  10.0.0.2, priority: 1, weight: 1, state: site-self, reachable
```

Step 6 **show lisp site** [**name** *site-name*]

The **show lisp site** command is useful for quickly verifying the operational status of LISP sites, as configured on a map server. This command only applies to a device configured as a map server. The following example output is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and shows the information for the instance ID of 101.

Example:

```
Router# show lisp site instance-id 101

LISP Site Registration Information

Site Name      Last      Up   Who Last             Inst     EID Prefix
               Register       Registered           ID
Left           00:00:36  yes  10.0.0.2             101      192.168.1.0/24
               00:00:43  yes  10.0.0.2             101      2001:DB8:A:A::/64
Right          00:00:31  yes  10.0.1.2             101      192.168.2.0/24
               00:00:02  yes  10.0.1.2             101      2001:DB8:A:B::/64
```

Example:

This second example shows LISP site information for the IPv6 EID prefix of 2001:db8:a:a:/64 and instance ID of 101.

```
Router# show lisp site 2001:db8:a:a:/64 instance-id 101

LISP Site Registration Information

Site name: Left
Allowed configured locators: any
Requested EID-prefix:
  EID-prefix: 2001:DB8:A:A::/64 instance-id 101
    First registered:     02:41:55
    Routing table tag:    0
    Origin:               Configuration
    Registration errors:
      Authentication failures:   4
      Allowed locators mismatch: 0
    ETR 10.0.0.2, last registered 00:00:22, no proxy-reply, no map-notify
                  TTL 1d00h
      Locator   Local  State      Pri/Wgt
      10.0.0.2  yes    up           1/1
```

Step 7 **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}

The LISP Internet Groper (lig) command is useful for testing the LISP control plane. The **lig** command can be used to query for the indicated destination hostname or EID, or the routers local EID-prefix. This command provides a simple means of testing whether a destination EID exists in the LISP mapping database system, or your site is registered with

the mapping database system. This command is applicable for both the IPv4 and IPv6 address families and applies to any IOS XE LISP device that maintains a map cache (for example, if configured as an ITR or PITR). The following example output is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and shows the information for the instance ID of 101 and the IPv4 EID prefix of 192.168.2.1.

Example:

```
Router# lig instance-id 101 192.168.2.1

Mapping information for EID 192.168.2.1 from 10.0.1.2 with RTT 12 msecs
192.168.2.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:00:00  up           1/1
```

Example:

This second example output shows information about the VRF named PURPLE:

```
Router# lig eid-table vrf PURPLE self

Mapping information for EID 192.168.1.0 from 10.0.0.1 with RTT 20 msecs
192.168.1.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.0.0.1  00:00:00  up, self     1/1
```

Step 8 **ping** {*hostname* | *destination-EID*}

The **ping** command is useful for testing basic network connectivity and reachability and/or liveness of a destination EID or RLOC address. When using **ping** it is important to be aware that because LISP uses an encapsulation, you should always specify a source address; never allow the **ping** application to assign its own default source address. This is because there are four possible ways to use **ping**, and without explicitly indicating the source address, the wrong one may be used by the application leading to erroneous results that complicate operational verification or troubleshooting. The four possible uses of **ping** include:

- RLOC-to-RLOC—Sends “echo?? packets out natively (no LISP encap) and receive the “echo-reply?? back natively. This can be used to test the underlying network connectivity between locators of various devices, such as xTR to Map-Server or Map-Resolver.
- EID-to-EID—Sends “echo?? packets out LISP-encaped and receive the “echo-reply?? back LISP-encaped. This can be used to test the LISP data plane (encapsulation) between LISP sites.
- EID-to-RLOC—Sends “echo?? packets out natively (no LISP encap) and receive the "echo-reply" back LISP-encaped through a PITR mechanism. This can be used to test the PITR infrastructure.
- RLOC-to-EID - Sends “echo?? packets out LISP-encaped and receive the “echo-reply?? back natively. This can be used to test PETR capabilities.

The **ping** command is applicable to the IPv4 and IPv6 address families respectively, and can be used on any IOS XE LISP device in some manner. (The ability to do LISP encapsulation, for example, requires the device to be configured as an ITR or PITR.)

The following example output from the **ping** command is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes. (Note that ping is not a LISP command and does not know about an EID table or an instance ID. When virtualization is included, output limiters can only be specified by VRF.)

Example:

```
Router# ping vrf PURPLE 2001:DB8:a:b::1 source 2001:DB8:a:a::1 rep 100
```

```
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 2001:DB8:A:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:A:A::1%PURPLE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 0/0/1 ms
```

Example:

```
Router# ping vrf GOLD

Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:b:b::1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: y
Source address or interface: 2001:db8:b:a::1
.
.
.
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:B:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:B:A::1%GOLD
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Step 9 **clear** [**ip** | **ipv6**] **lisp map-cache**

The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries stored by the router. This can be useful trying to quickly verify the operational status of the LISP control plane. This command applies to a LISP device that maintains a map cache (for example, if configured as an ITR or PITR).

Example:

The following example displays IPv4 mapping cache information for instance ID 101, shows the command used to clear the mapping cache for instance ID 101, and displays the show information after clearing the cache.

```
Router# show ip lisp map-cache instance-id 101

LISP IPv4 Mapping Cache for EID-table vrf PURPLE (IID 101), 2 entries

0.0.0.0/0, uptime: 00:25:17, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
192.168.2.0/24, uptime: 00:20:13, expires: 23:39:39, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:20:13  up           1/1

Router# clear ip lisp map-cache instance-id 101

Router# show ip lisp map-cache instance-id 101

LISP IPv4 Mapping Cache, 1 entries

0.0.0.0/0, uptime: 00:00:02, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

Configuration Examples for LISP Parallel Model Virtualization

Complete configuration examples are available within each task under the “How to Configure LISP Parallel Model Virtualization” section.

Additional References

Related Documents

Document Title	Location
Cisco IOS IP Routing: LISP Command Reference	http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_lisp/command/ip-lisp-cr-book.html
Enterprise IPv6 Transitions Strategy Using the Locator/ID Separation Protocol	Cisco LISP Software Image Download Page
Cisco IOS LISP0 Virtual Interface, Application Note, Version 1.0	Cisco LISP Software Image Download Page
Cross-Platform Release Notes for Cisco IOS Release 15.2M&T	http://www.cisco.com/en/US/docs/ios/15_2m_and_t/release/notes/15_2m_and_t.html

Standards

Standard	Title
IANA Address Family Numbers	http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xml

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
draft-ietf-lisp-22	Locator/ID Separation Protocol (LISP) http://tools.ietf.org/html/draft-ietf-lisp-22
draft-ietf-lisp-ms-16	LISP Map Server http://tools.ietf.org/html/draft-ietf-lisp-ms-16
draft-ietf-lisp-alt-10	LISP Alternative Topology (LISP+ALT) http://tools.ietf.org/html/draft-ietf-lisp-alt-10
draft-ietf-lisp-LCAF-06	LISP Canonical Address Format (LCAF) http://tools.ietf.org/wg/lisp/
draft-ietf-lisp-interworking-06	Interworking LISP with IPv4 and IPv6 http://tools.ietf.org/html/draft-ietf-lisp-interworking-06
draft-ietf-lisp-lig-06	LISP Internet Groper (LIG) http://tools.ietf.org/html/draft-ietf-lisp-lig-06
draft-ietf-lisp-mib-03	LISP MIB http://tools.ietf.org/wg/lisp/draft-ietf-lisp-mib/

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Parallel Model Virtualization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 195: Feature Information for LISP Parallel Model Virtualization

Feature Name	Releases	Feature Information
LISP Parallel Model Virtualization	15.2(3)T	LISP Parallel Model Virtualization ties virtualized EID space associated with VRFs to RLOCs associated with the same or different VRFs.

CHAPTER 182

LISP Host Mobility Across Subnet

Information About LISP Host Mobility Across Subnet

Devices configured with LISP Host Mobility ASM have the following characteristics:

- Each edge router (xTR) is the first Layer-3 hop
- Proxy-arp is enabled on the xTR's gateway interface
- Each roaming site xTR should register with a common set of map-servers
- Mobility hosts should not be "silent" after they move
- A multicast configuration is needed by xTRs only if the site has multiple xTRs, for example for HSRP. A single xTR does not need to use multicasting.
- Supports vmotion or live host mobility only in the case of North-South traffic
- LISP encapsulation (ASM) is required for East-West traffic.

Overview of LISP Host Mobility Across Subnet

You can use LISP Host Mobility Across Subnet commands to deploy extended subnets and across subnets. A detailed configuration guide and examples are under development and will appear here soon. Meanwhile, please refer to the LISP Command Reference.

CHAPTER 183

LISP Delegate Database Tree (DDT)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Delegate Database Tree (DDT)

Overview of LISP Delegate Database Tree (DDT)

You can use LISP Delegate Database Tree (DDT) commands to deploy a distributed LISP Mapping System. A detailed configuration guide and examples are under development and will appear here soon. Meanwhile, please refer to the *LISP Command Reference*.

CHAPTER 184

LISP ESM Multihop Mobility

The LISP ESM Multihop Mobility feature separates the Locator/ID Separation Protocol (LISP) dynamic host detection function from the LISP encapsulation/decapsulation function within a LISP topology.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for LISP ESM Multihop Mobility

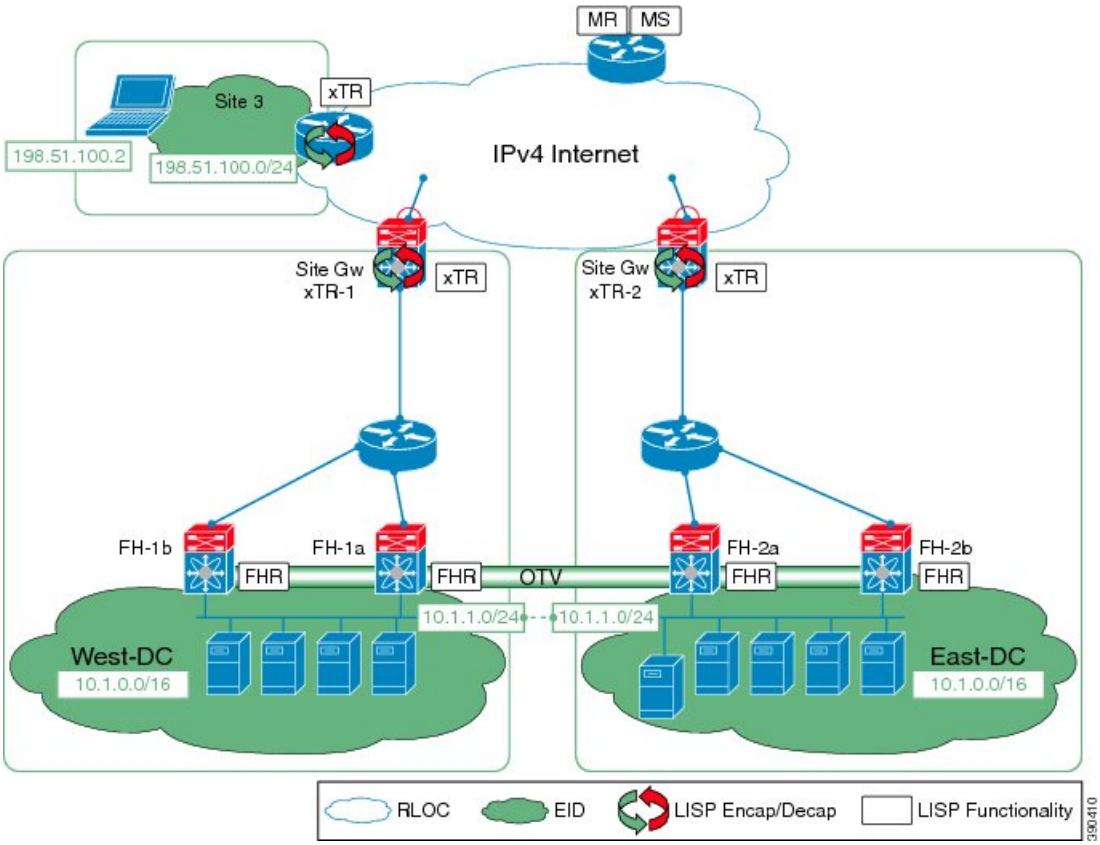
- Supports Locator/ID Separation Protocol (LISP) multihop mobility only in Extended Subnet Mode (ESM) with Overlay Transport Virtualization (OTV).
- Requires OTV First Hop Redundancy Protocol (FHRP) isolation to avoid hair-pinning of traffic across the OTV Data Center Interconnect (DCI) framework.
- Does not support Network Address Translated (NAT'd) endpoint identifiers (EIDs).

Information About LISP ESM Multihop Mobility

LISP ESM Multihop Mobility Overview

A first-hop router (FHR) detects the presence of a dynamic host endpoint identifier (EID) and notifies the site gateway xTR. A device configured as both an ingress tunnel router (ITR) and an egress tunnel router (ETR) is known as an xTR. The site gateway xTR registers the dynamic EID with a map server. The Site Gateway xTR performs Locator/ID Separation Protocol (LISP) encapsulation/decapsulation of the traffic from or to the dynamic EID to or from remote sites.

Figure 189: LISP ESM Multihop Mobility Sample Topology



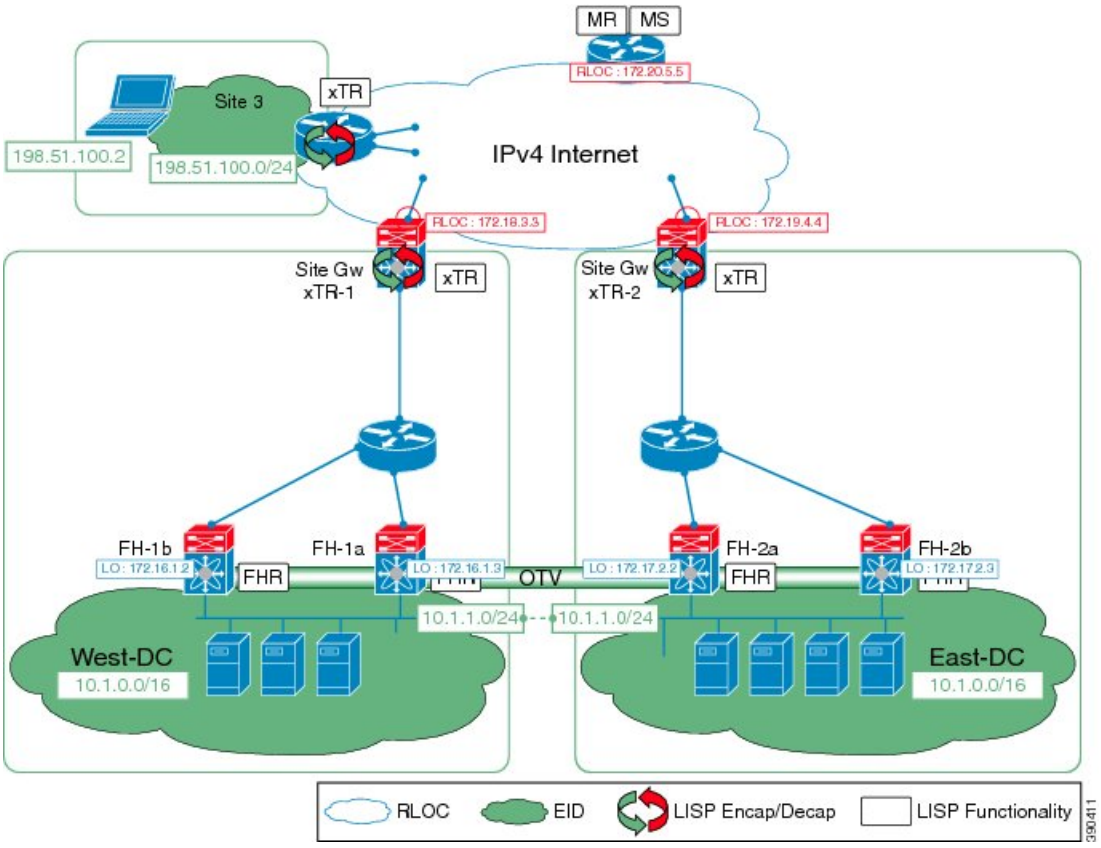
Multiple Layer 3 hops can exist between the FHR and the site gateway xTR when deploying the LISP ESM Multihop Mobility feature. You can insert non-LISP devices like firewalls and load-balancers into the data center.

Note LISP supports silent host moves from the 15.4(1)T release.

Note LISP supports redistributing host routes for servers discovered by LISP into Interior Gateway Protocol (IGP) via Open Shortest Path First (OSPF) protocol/ Intermediate System-to-Intermediate System (IS-IS) protocol/ Routing Information Protocol (RIP)/ Border Gateway Protocol (BGP).

Perform the tasks shown below to configure LISP ESM multihop mobility on a Locator ID/Separation Protocol (LISP) site with three IPv4 routing locators (RLOCs). In these tasks, a LISP site uses a single edge router configured as both an ITR and an ETR (known as an xTR) with two connections to the upstream provider. Both the RLOCs and the endpoint identifier (EID) prefix are IPv4. The LISP site registers to a map resolver map server (MRMS) device in the network core. The topology used in this LISP configuration is shown in the figure below.

Figure 190: Topology for LISP ESM Multihop Mobility



The components illustrated in the topology shown in the above figure are described below:

LISP Site

- The customer premises equipment (CPE) functions as a LISP ITR and ETR (xTR).
- The LISP xTR is authoritative for the IPv4 EID prefix of 10.1.0.0/16.
- The LISP xTR has two RLOC connections to the core. The RLOC connection to xTR-1 is 172.18.3.3; the RLOC connection to xTR-2 is 172.19.4.4.

Mapping System

- An MRMS system is assumed to be available for the LISP xTRs to configure. The MRMS has IPv4 RLOCs 10.1.1.0 and 10.1.1.9.
- Mapping services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system.

How to Configure LISP ESM Multihop Mobility

Configuring First-Hop Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **locator-set** *locator-set-name*
5. *ipv4-address* **priority** *priority-locator* **weight** *locator-weight*
6. Repeat Step 5 to configure another locator entry.
7. **exit**
8. **eid-table default instance-id** *id*
9. **dynamic-eid** *dynamic-eid-name*
10. **database-mapping** *dynamic-eid-prefix*/`prefix-length` **locator-set** *name*
11. **eid-notify** *ipv4-address* **key** *password*
12. **map-notify-group** *ipv4-group-address*
13. **exit**
14. **exit**
15. **exit**
16. **interface** *type number*
17. **lisp mobility** *dynamic-eid-name*
18. **lisp extended-subnet-mode**
19. **ip address** *ip-address mask*
20. **standby** *group-number* **ip** *virtual-ip-address*
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**locator-set** *locator-set-name* **Example:** `Device(config-router-lisp)# locator-set WestDC`	Specifies a locator set and enters LISP locator-set configuration mode.
Step 5	*ipv4-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 172.16.1.2 priority 10 weight 50`	Configures the LISP locator set. The LISP locator set is the set of addresses that the first-hop router (FHR) uses while communicating with the gateway xTR. You can configure each locator address by creating a locator entry with an assigned priority and weight.
Step 6	Repeat Step 5 to configure another locator entry.	—
Step 7	**exit** **Example:** `Device(config-router-lisp-locator-set)# exit`	Exits LISP locator-set configuration mode and returns to LISP configuration mode.
Step 8	**eid-table default instance-id** *id* **Example:** `Device(config-router-lisp)# eid-table default instance-id 0`	Configures an association between the default virtual routing and forwarding (VRF) table and a LISP instance ID, and enters EID table configuration mode.
Step 9	**dynamic-eid** *dynamic-eid-name* **Example:** `Device(config-router-lisp-eid-table)# dynamic-eid VMs`	Specifies a LISP virtual machine (VM)-mobility (dynamic EID roaming) policy and enters dynamic EID configuration mode.
Step 10	**database-mapping** *dynamic-eid-prefix*/`prefix-length` **locator-set** *name* **Example:** `Device(config-router-lisp-eid-table-dynamic-eid)# database-mapping 10.1.1.0/24 locator-set WestDC`	Configures an IPv4 mapping relationship and an associated traffic policy for the LISP VM-mobility (dynamic EID) policy. **Note** You can enter the **limit dynamic** *value* keyword to limit the number of discoverable dynamic EIDs. However, if you have enabled debug mode (using the **service internal** command), then the number of discoverable dynamic EIDs will be increased to a fixed value of 65535.
Step 11	**eid-notify** *ipv4-address* **key** *password* **Example:** `Device(config-router-lisp-eid-table-dynamic-eid)# eid-notify 192.0.2.21 key k`	Enables sending of dynamic endpoint identifier (EID) presence notifications to a gateway xTR with the specified IPv4 address along with the authentication key used with the gateway xTR.

	Command or Action	Purpose
Step 12	**map-notify-group** *ipv4-group-address* **Example:** `Device(config-router-lisp-eid-table-dynamic-eid)# map-notify-group 224.0.0.0`	Specifies the IPv4 multicast group address used for sending and receiving site-based map-notify multicast messages.
Step 13	**exit** **Example:** `Device(config-router-lisp-eid-table-dynamic-eid)# exit`	Exits dynamic EID configuration mode and returns to EID table configuration mode.
Step 14	**exit** **Example:** `Device(config-router-lisp-eid-table)# exit`	Exits EID table configuration mode and returns to LISP configuration mode.
Step 15	**exit** **Example:** `Device(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 16	**interface** *type number* **Example:** `Device(config)# interface Vlan 11`	Specifies the interface type and number and enters interface configuration mode.
Step 17	**lisp mobility** *dynamic-eid-name* **Example:** `Device(config-if)# lisp mobility VMs`	Allows EID mobility on the interface and specifies the name of the dynamic EID.
Step 18	**lisp extended-subnet-mode** **Example:** `Device(config-if)# lisp extended-subnet-mode`	Enables extended subnet mode on the interface.
Step 19	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.2 255.255.255.0`	Configures an IPv4 address for a specific interface.
Step 20	**standby** *group-number* **ip** *virtual-ip-address* **Example:** `Device(config-if)# standby 1 ip 10.1.1.1`	Enables IPv4 Hot Standby Router Protocol (HSRP) and sets the virtual IP address.
Step 21	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring Site Gateway xTR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **locator-set** *locator-set-name*
5. *ipv4-address* **priority** *priority-locator* **weight** *locator-weight*
6. **exit**
7. **eid-table default instance-id** *id*
8. **database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name*
9. **dynamic-eid** *dynamic-eid-name*
10. **database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name*
11. **eid-notify authentication-key** *password*
12. **exit**
13. **exit**
14. **ipv4 itr map-resolver** *map-resolver-address*
15. **ipv4 itr**
16. **ipv4 etr map-server** *map-server-address* **key** *authentication-key*
17. **ipv4 etr**
18. **exit**
19. **interface** *type number*
20. **ip address** *ip-address mask*
21. **lisp mobility** *dynamic-eid-name*
22. **lisp extended-subnet-mode**
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**locator-set** *locator-set-name* **Example:**	Specifies a locator set and enters LISP locator-set configuration mode.

	Command or Action	Purpose
	`Device(config-router-lisp)# locator-set WestDC`	
Step 5	*ipv4-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 172.18.3.3 priority 10 weight 50`	Configures the LISP locator set. The LISP locator set is the set of addresses used by the gateway xTR while encapsulating/decapsulating LISP traffic from and to the endpoint identifier (EID).
Step 6	**exit** **Example:** `Device(config-router-lisp-locator-set)# exit`	Exits LISP locator-set configuration mode and returns to LISP configuration mode.
Step 7	**eid-table default instance-id** *id* **Example:** `Device(config-router-lisp)# eid-table default instance-id 0`	Configures an association between the default virtual routing and forwarding (VRF) table and a LISP instance ID, and enters EID table configuration mode.
Step 8	**database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name* **Example:** `Device(config-router-lisp-eid-table)# database-mapping 10.1.0.0/16 locator-set WestDC`	Configures an IPv4 mapping relationship and an associated traffic policy for LISP virtual machine (VM)-mobility (dynamic EID) policy. **Note** You can enter the **limit dynamic** *value* keyword to limit the number of discoverable dynamic EIDs. However, if you have enabled debug mode (using the **service internal** command), then the number of discoverable dynamic EIDs will be increased to a fixed value of 65535.
Step 9	**dynamic-eid** *dynamic-eid-name* **Example:** `Device(config-router-lisp-eid-table)# dynamic-eid VMs`	Specifies a LISP VM-mobility (dynamic EID roaming) policy and enters dynamic EID configuration mode.
Step 10	**database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name* **Example:** `Device(config-router-lisp-eid-table-dynamic-eid)# database-mapping 10.1.1.0/24 locator-set WestDC`	Configures an IPv4 mapping relationship and an associated traffic policy for LISP VM-mobility (dynamic EID) policy.
Step 11	**eid-notify authentication-key** *password* **Example:** `Device(config-router-lisp-eid-table-dynamic-eid)# eid-notify authentication-key k`	Specifies the authentication key to validate the EID-notify sent from a first-hop router (FHR).
Step 12	**exit** **Example:** `Device(config-router-lisp-eid-table-dynamic-eid)# exit`	Exits dynamic EID configuration mode and returns to EID table configuration mode.

	Command or Action	Purpose
Step 13	**exit** **Example:** `Device(config-router-lisp-eid-table)# exit`	Exits EID table configuration mode and returns to LISP configuration mode.
Step 14	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Device(config-router-lisp)# ipv4 itr map-resolver 172.20.5.5`	Configures a locator address for the LISP map resolver to which this device will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. **Note** You can configure up to 8 map resolvers if multiple map resolvers are available.
Step 15	**ipv4 itr** **Example:** `Device(config-router-lisp)# ipv4 itr`	Enables LISP ingress tunnel router (ITR) functionality for the IPv4 address family.
Step 16	**ipv4 etr map-server** *map-server-address* **key** *authentication-key* **Example:** `Device(config-router-lisp)# ipv4 etr map-server 172.20.5.5 key mskey`	Configures the IPv4 or IPv6 locator address of the LISP map server to be used by the egress tunnel router (ETR) when registering IPv4 endpoint identifiers (EIDs).
Step 17	**ipv4 etr** **Example:** `Device(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 18	**exit** **Example:** `Device(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 19	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/4`	Specifies the interface type and number and enters interface configuration mode.
Step 20	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.2.21 255.255.255.0`	Configures an IPv4 address for the interface.
Step 21	**lisp mobility** *dynamic-eid-name* **Example:** `Device(config-if)# lisp mobility VMs`	Allows EID mobility on the interface and specifies the name of the dynamic EID.
Step 22	**lisp extended-subnet-mode** **Example:**	Enables extended subnet mode on the interface.

	Command or Action	Purpose
	`Device(config-if)# lisp extended-subnet-mode`	
Step 23	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring xTR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **locator-set** *locator-set-name*
5. *ipv4-address* **priority** *priority-locator* **weight** *locator-weight*
6. Repeat Step 5 to configure another locator entry.
7. **exit**
8. **eid-table default instance-id** *id*
9. **database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name*
10. **exit**
11. **ipv4 itr map-resolver** *map-resolver-address*
12. **ipv4 itr**
13. **ipv4 etr map-server** *map-server-address* **key** *authentication-key*
14. **ipv4 etr**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**locator-set** *locator-set-name* **Example:**	Specifies a locator set and enters LISP locator-set configuration mode.

	Command or Action	Purpose
	`Device(config-router-lisp)# locator-set Site3RLOCS`	
Step 5	*ipv4-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 203.0.113.2 priority 10 weight 50`	Configures the LISP locator set. The LISP locator set is the set of addresses used by the gateway xTR while encapsulating/decapsulating LISP traffic from and to the endpoint identifier (EID).
Step 6	Repeat Step 5 to configure another locator entry.	—
Step 7	**exit** **Example:** `Device(config-router-lisp-locator-set)# exit`	Exits LISP locator set configuration mode and returns to LISP configuration mode.
Step 8	**eid-table default instance-id** *id* **Example:** `Device(config-router-lisp)# eid-table default instance-id 0`	Configures an association between the default VRF table and a LISP instance ID, and enters EID table configuration mode.
Step 9	**database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name* **Example:** `Device(config-router-lisp-eid-table)# database-mapping 198.51.100.0/24 locator-set Site3RLOCS`	Configures an IPv4 mapping relationship and an associated traffic policy for the LISP Virtual Machine (VM)-mobility (dynamic EID) policy. **Note** You can enter the **limit dynamic** *value* keyword to limit the number of discoverable dynamic EIDs. However, if you have enabled debug mode (using the **service internal** command), then the number of discoverable dynamic EIDs will be increased to a fixed value of 65535.
Step 10	**exit** **Example:** `Device(config-router-lisp-eid-table)# exit`	Exits EID table configuration mode and returns to LISP configuration mode.
Step 11	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Device(config-router-lisp)# ipv4 itr map-resolver 172.20.5.5`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. **Note** You can configure up to 8 map resolvers if multiple map resolvers are available.
Step 12	**ipv4 itr** **Example:** `Device(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for an IPv4 address family.

	Command or Action	Purpose
Step 13	**ipv4 etr map-server** *map-server-address* **key** *authentication-key* **Example:** `Device(config-router-lisp)# ipv4 etr map-server 172.20.5.5 key k3`	Configures IPv4 locator address of the LISP map server to be used by the egress tunnel router (ETR) when registering for IPv4 endpoint identifiers (EIDs).
Step 14	**ipv4 etr** **Example:** `Device(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for an IPv4 address family.
Step 15	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring Map Server Map Resolver

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **site** *site-name*
5. **authentication-key** *password*
6. **eid-prefix** *eid-prefix* **accept-more-specifics**
7. **exit**
8. Repeat Step 4 to Step 7 to configure another LISP site.
9. **ipv4 map-server**
10. **ipv4 map-resolver**
11. **end**

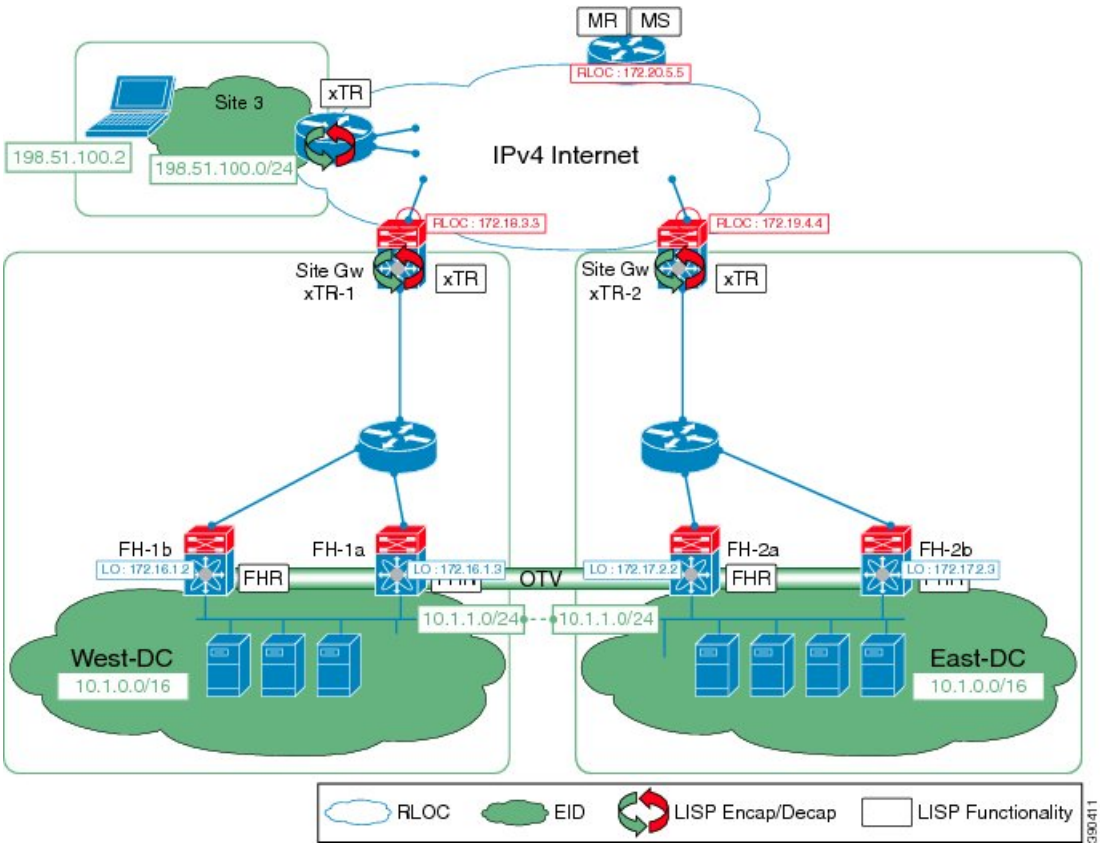
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:**	Enters Locator ID/Separation Protocol (LISP) configuration mode.

	Command or Action	Purpose
	`Device(config)# router lisp`	
Step 4	**site** *site-name* **Example:** `Device(config-router-lisp)# site EastWestDC`	Configures a LISP site and enters LISP site configuration mode on a LISP map server.
Step 5	**authentication-key** *password* **Example:** `Device(config-router-lisp-site)# authentication-key k`	Configures the password used to create the Hash-based Message Authentication Code (HMAC) Secure Hash Algorithm (SHA-1) hash for authenticating the map-register message sent by an egress tunnel router (ETR) when registering with the map server.
Step 6	**eid-prefix** *eid-prefix* **accept-more-specifics** **Example:** `Device(config-router-lisp-site)# eid-prefix 10.1.0.0/16 accept-more-specifics`	Configures a list of endpoint identifier (EID) prefixes that are allowed in a map-register message sent by an ETR when registering with the map server. Specifies that any EID prefix that is more specific than the EID prefix configured is accepted and tracked.
Step 7	**exit** **Example:** `Device(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 8	Repeat Step 4 to Step 7 to configure another LISP site.	—
Step 9	**ipv4 map-server** **Example:** `Device(config-router-lisp)# ipv4 map-server`	Configures a device to act as an IPv4 LISP map server.
Step 10	**ipv4 map-resolver** **Example:** `Device(config-router-lisp)# ipv4 map-resolver`	Configures a device to act as an IPv4 LISP map resolver.
Step 11	**end** **Example:** `Device(config-router-lisp)# end`	Exits LISP configuration mode and returns to privileged EXEC mode.

Configuration Examples for LISP ESM Multihop Mobility

Figure 191: LISP ESM Multihop Topology



The examples below show the complete configuration for the LISP topology illustrated in the figure above.

Example: First-Hop Router Configuration

```
Device# configure terminal
Device(config)# router lisp
Device(config-router-lisp)# locator-set WestDC
Device(config-router-lisp-locator-set)# 172.16.1.2 priority 10 weight 50
Device(config-router-lisp-locator-set)# 172.17.2.3 priority 10 weight 50
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# eid-table default instance-id 0
Device(config-router-lisp-eid-table)# dynamic-eid VMs
Device(config-router-lisp-eid-table-dynamic-eid)# database-mapping 10.1.1.0/24 locator-set
 WestDC
Device(config-router-lisp-eid-table-dynamic-eid)# eid-notify 192.0.2.21 key k
Device(config-router-lisp-eid-table-dynamic-eid)# map-notify-group 224.0.0.0
Device(config-router-lisp-eid-table-dynamic-eid)# exit
Device(config-router-lisp-eid-table)# exit
Device(config-router-lisp)# exit
Device(config)# interface Vlan11
Device(config-if)# lisp mobility VMs
```

```
Device(config-if)# lisp extended-subnet-mode
Device(config-if)# ip address 10.1.1.2 255.255.255.0
Device(config-if)# standby 1 ip 10.1.1.1
```

Example: Site Gateway xTR Configuration

```
Device> enable
Device# configure terminal
Device (config)# router lisp
Device(config-router-lisp)# locator-set WestDC
Device(config-router-lisp-locator-set) # 172.18.3.3 priority 10 weight 50
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# eid-table default instance-id 0
Device(config-router-lisp-eid-table)# database-mapping 10.1.0.0/16 locator-set WestDC
Device(config-router-lisp-eid-table)# dynamic-eid VMs
Device(config-router-lisp-eid-table-dynamic-eid)# database-mapping 10.1.1.0/24 locator-set
 WestDC
Device(config-router-lisp-eid-table-dynamic-eid)# eid-notify authentication-key k
Device(config-router-lisp-eid-table-dynamic-eid)# exit
Device(config-router-lisp-eid-table)# exit
Device(config-router-lisp)# ipv4 itr map-resolver 172.20.5.5
Device(config-router-lisp)# ipv4 itr
Device(config-router-lisp)# ipv4 etr map-server 172.20.5.5 key k
Device(config-router-lisp)# ipv4 etr
Device(config-router-lisp)# exit
Device(config)# interface FastEthernet1/4
Device(config-if)# ip address 192.0.2.21 255.255.255.0
Device(config-if)# lisp mobility VMs
Device(config-if)# lisp extended-subnet-mode
```

Example: xTR Configuration

```
Device> enable
Device# configure terminal
Device (config)# router lisp
Device(config-router-lisp)# locator-set Site3RLOCS
Device(config-router-lisp-locator-set)# 203.0.113.2 priority 10 weight 50
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# eid-table default instance-id 0
Device(config-router-lisp-eid-table)# database-mapping 198.51.100.0/24 locator-set Site3RLOCS
Device(config-router-lisp-eid-table)# exit
Device(config-router-lisp)# ipv4 itr map-resolver 172.20.5.5
Device(config-router-lisp)# ipv4 itr
Device(config-router-lisp)# ipv4 etr map-server 172.20.5.5 key k3
Device(config-router-lisp)# ipv4 etr
```

Example: Map Server Map Resolver Configuration

```
Device> enable
Device# configure terminal
Device (config)# router lisp
Device(config-router-lisp)# site EastWestDC
Device(config-router-lisp-site)# authentication-key k
Device(config-router-lisp-site)# eid-prefix 10.1.0.0/16 accept-more-specifics
Device(config-router-lisp-site)# exit
```

```
Device(config-router-lisp)# ipv4 map-server
Device(config-router-lisp)# ipv4 map-resolver
```

Additional References for LISP ESM Multihop Mobility

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Locator/ID Separation Protocol (LISP) commands	Cisco IOS IP Routing: LISP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP ESM Multihop Mobility

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Release	Feature Information
LISP ESM Multihop Mobility		The LISP ESM Multihop Mobility feature separates the Locator/ID Separation Protocol (LISP) dynamic host detection function from the LISP encapsulation/decapsulation function within a LISP topology.

CHAPTER 185

LISP Support for Disjoint RLOC Domains

The Locator/ID Separation Protocol (LISP) implements a “level of indirection” that enables a new IP routing architecture. LISP separates IP addresses into two namespaces: Endpoint Identifiers (EIDs), which are assigned to end-hosts, and Routing Locators (RLOCs), which are assigned to devices that make up the global routing system.

The LISP Support for Disjoint RLOC Domains feature enables LISP-to-LISP communication between LISP sites that are connected to different RLOC spaces but have no connectivity to each other. One example of disjointed RLOC space is that of between the IPv4 Internet and IPv6 Internet. When one LISP site has IPv4-only RLOC connectivity and the second site has IPv6-only RLOC connectivity, these sites can still communicate via LISP using the LISP Support for Disjoint RLOC Domains feature.

Prerequisites for LISP Support for Disjoint RLOC Domains

Map servers and re-encapsulating tunnel routers (RTRs) must have connectivity to all locator spaces that are being joined.

Restrictions for LISP Support for Disjoint RLOC Domains

Map servers and re-encapsulating tunnel routers (RTRs) cannot join more than eight locator scopes.

Information About LISP Support for Disjoint RLOC Domains

LISP Support for Disjoint RLOC Domains Overview

The fundamental principal of any network is that routing and reachability must exist between all devices that make up the total network system. There are many network systems, public and private, for which internetwork connectivity is not directly available. A few examples include:

- IPv4 Internet and IPv6 Internet.
- An IPv4 Multiprotocol Label Switching (MPLS) VPN from service provider A and an IPv4 MPLS VPN from service provider B.
- An IPv4 MPLS VPN from service provider A and IPv4 Internet.

When some sites within a network connect to one routing domain and other sites connect to another routing domain, a gateway function must be provided to facilitate connectivity between these disjointed routing domains. In traditional routing architectures, providing connectivity between disjointed routing domains can be quite complex.

The inherent property of Locator/ID Separation Protocol (LISP), which separates IP addresses into two namespaces, endpoint identifiers (EIDs) and routing locators (RLOCs), also gives it the ability to connect disjointed RLOC domains. The LISP Support for Disjoint RLOC Domains feature provides simplified configuration mechanisms that enable this capability. The key components are new control plane configuration options on the LISP map server, and a functionality called re-encapsulating tunnel router (RTR), which provides data plane connectivity between disjointed locator spaces.

LISP Map Server

The key concept in the LISP Support for Disjoint RLOC Domains feature is the recognition that the LISP Mapping System has full knowledge of all LISP sites. When a LISP site registers with a map server, the registration message not only provides information about the EID space that the site is authoritative for, but it also provides information about its own RLOCs.

The LISP Support for Disjoint RLOC Domains feature provides new configuration options to define within the map server the routing locator scopes that LISP sites can connect to. Once defined, the map server automatically determines whether individual sites have common or disjoint locator connectivity between themselves. The map server then uses this knowledge when handling Map-Request messages to determine how to inform LISP sites to communicate with each other. Map-Request messages contain both source and destination EID information. When a map server receives a Map-Request message, it compares the RLOCs associated with the source EID and destination EID contained with the Map-Request message against the configured locator scopes.

- If the ingress tunnel router (ITR) (source EID) and egress tunnel router (ETR) (destination EID) share at least one RLOC in a common locator scope, the map server forwards the Map-Request message to the ETR as normal. In this case, the ETR is capable of generating a Map-Reply message that is sent back to the ITR since it has reachability across (at least one) common locator space.
- If the ITR (source EID) and ETR (destination EID) do not share at least one RLOC in a common locator scope, the map server sends a proxy Map-Reply message to the ITR that includes a list of RTRs that are capable of connecting the disjointed locator space between the ITR and ETR.
- If the RLOCs associated with the ITR (source EID) and ETR (destination EID) do not match any configured locator scopes, the map server forwards the Map-Request message to the ETR as normal. In

this case, the RLOCs are assumed to be reachable via routing, even though they are not defined in any locator scope configuration.

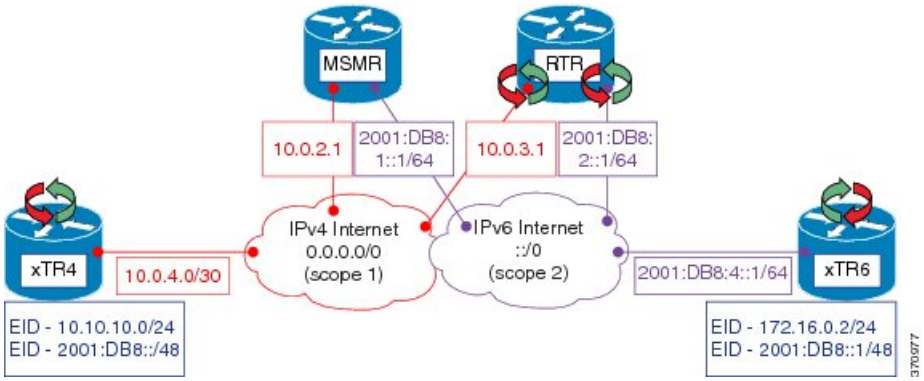
LISP data plane packets flow directly between sites when the sites share locator space. An RTR is used to connect LISP data plane packets when locator spaces between the sites are disjointed.

LISP RTR

A re-encapsulating tunnel router (RTR) provides data plane communications support for LISP-to-LISP traffic between LISP sites that do not share common locator space. Functionally, an RTR takes in LISP encapsulated packets from an ITR in one locator scope, decapsulates them, does a map-cache lookup, and then re-encapsulates them to an ETR in another locator scope. The following are important considerations for an RTR:

- The RTR itself must have RLOCs in all locator scopes that are being joined.
- An RTR sends Map-Request messages to populate its own map cache. As a Map-Request message contains an ITR RLOC field that is populated with one or more entries corresponding to the locators of the device sending the Map-Request message, the RTR in this case, the locator set configuration is also required on the RTR to define its locators. This enables the map server to correctly receive Map-Requests from the RTR to assess locator scope connectivity.
- An RTR performs functions similar to a proxy ingress tunnel router (PITR) and proxy egress tunnel router (PETR), therefore these features must be enabled on the RTR.

Figure 192: LISP - Disjoint RLOC Domains Topology



Referring to Figure 1, the tasks below illustrate the configuration steps required to provide Locator/ID Separation Protocol (LISP) Disjoint Routing Locator (RLOC) support for cross address-family (IPv4/IPv6) connectivity.

- Ingress/Egress tunnel router (xTR) represents the LISP Site router. In Figure 1, xTR4 only has RLOC connectivity to the IPv4 Internet, and xTR6 only has RLOC connectivity to the IPv6 Internet.
- Map server map resolver (MSMR) represents the MSMR supporting the LISP control plane.
- Re-encapsulating tunnel router (RTR) represents the LISP data plane device that joins locator scopes.

How to configure LISP Support for Disjoint RLOC Domains

Configuring xTR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ipv6 address** *ipv6-address/ipv6-prefix*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **router lisp**
9. **locator-set** *locator-set-name*
10. *ipv4-address* **priority** *priority-locator* **weight** *locator-weight*
11. *ipv6-address* **priority** *priority-locator* **weight** *locator-weight*
12. **exit**
13. **eid-table default instance-id** *id*
14. **database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name*
15. **database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name*
16. **exit**
17. **ipv4 itr map-resolver** *map-resolver-address*
18. **ipv4 itr**
19. **ipv4 etr map-server** *map-server-address* **key** *authentication-key*
20. **ipv4 etr**
21. **ipv6 itr map-resolver** *map-resolver-address*
22. **ipv6 itr**
23. **ipv6 etr map-server** *map-server-address* **key** *authentication-key*
24. **ipv6 etr**
25. **exit**
26. **ip route** *prefix mask ip-address*
27. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface loopback0`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.10.10.4 255.255.255.0`	Configures an IPv4 address for the interface.
Step 5	**ipv6 address** *ipv6-address/ipv6-prefix* **Example:** `Device(config-if)# ipv6 address 2001:DB8:0:ABCD::1/64`	Configures an IPv6 address for the interface.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.4.1 255.255.255.252`	Configures an IPv4 address for the interface.
Step 8	**router lisp** **Example:** `Device(config-if)# router lisp`	Enters LISP configuration mode.
Step 9	**locator-set** *locator-set-name* **Example:** `Device(config-router-lisp)# locator-set R4`	Specifies a locator set and enters LISP locator set configuration mode.
Step 10	*ipv4-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 10.0.4.1 priority 1 weight 1`	Configures the LISP locator set. The LISP locator set is the set of addresses the first-hop router uses when communicating with the gateway xTR. You can configure each IPv4 locator address by creating a locator entry with assigned priority and weight.
Step 11	*ipv6-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 2001:DB8:4::2 priority 1 weight 1`	Configures the LISP locator set. The LISP locator set is the set of addresses the first-hop router uses when communicating with the gateway xTR. You can configure each IPv6 locator address by creating a locator entry with assigned priority and weight.
Step 12	**exit** **Example:**	Exits LISP locator set configuration mode and returns to LISP configuration mode.

	Command or Action	Purpose
	`Device(config-router-lisp-locator-set)# exit`	
Step 13	**eid-table default instance-id** *id* **Example:** `Device(config-router-lisp)# eid-table default instance-id 0`	Configures an association between the default (global) routing table and a LISP instance ID, and enters EID table configuration mode.
Step 14	**database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name* **Example:** `Device(config-router-lisp-eid-table)# database-mapping 10.10.10.0/24 locator-set R4`	Configures an IPv4/IPv6 mapping relationship and an associated traffic policy (as defined in the locator set) for this LISP site.
Step 15	**database-mapping** *dynamic-eid-prefix/prefix-length* **locator-set** *name* **Example:** `Device(config-router-lisp-eid-table)# database-mapping 2001:DB8::/48 locator-set R4`	Configures an IPv4/IPv6 mapping relationship and an associated traffic policy (as defined in the locator set) for this LISP site.
Step 16	**exit** **Example:** `Device(config-router-lisp-eid-table)# exit`	Exits EID table configuration mode and returns to LISP configuration mode.
Step 17	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Device(config-router-lisp)# ipv4 itr map-resolver 10.0.2.1`	Configures a locator address for the LISP map resolver to which this device will send Map-Request messages for IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. **Note** You can configure up to eight map resolvers if multiple map resolvers are available.
Step 18	**ipv4 itr** **Example:** `Device(config-router-lisp)# ipv4 itr`	Enables LISP ingress tunnel router (ITR) functionality for an IPv4 address family.
Step 19	**ipv4 etr map-server** *map-server-address* **key** *authentication-key* **Example:** `Device(config-router-lisp)# ipv4 etr map-server 10.0.2.1 key R4KEY`	Configures the IPv4 locator address of the LISP map server to be used by the egress tunnel router (ETR) when registering itself for IPv4 endpoint identifiers (EIDs).
Step 20	**ipv4 etr** **Example:** `Device(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for an IPv4 address family.

	Command or Action	Purpose
Step 21	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Device(config-router-lisp)# ipv6 itr map-resolver 10.0.2.1`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. **Note** You can configure up to eight map resolvers if multiple map resolvers are available.
Step 22	**ipv6 itr** **Example:** `Device(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for an IPv6 address family.
Step 23	**ipv6 etr map-server** *map-server-address* **key** *authentication-key* **Example:** `Device(config-router-lisp)# ipv6 etr map-server 10.0.2.1 key R4KEY`	Configures the IPv6 locator address for the LISP map server to be used by the ETR when registering for IPv6 EIDs.
Step 24	**ipv6 etr** **Example:** `Device(config-router-lisp)# ipv6 etr`	Enables LISP ETR functionality for an IPv6 address family.
Step 25	**exit** **Example:** `Device(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 26	**ip route** *prefix mask ip-address* **Example:** `Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.4.2`	Establishes static routes to the next hop destination.
Step 27	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring MSMR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ipv6 address** *ipv6-address/ipv6-prefix*

6. **router lisp**
7. **locator-set** *locator-set-name*
8. *ipv4-address* **priority** *priority-locator* **weight** *locator-weight*
9. **exit**
10. Repeat Step 7 to Step 9 to specify and configure another locator set.
11. **locator-scope** *name*
12. **rtr-locator-set** *locator-set-name*
13. **rloc-prefix** *ipv4-rloc-prefix*
14. **exit**
15. Repeat Step 11 to Step 14 to specify and configure another locator scope.
16. **site** *site-name*
17. **authentication-key** *password*
18. **eid-prefix** *ipv4-eid-prefix*
19. **eid-prefix** *ipv6-eid-prefix*
20. **exit**
21. Repeat Step 16 to Step 20 to configure another LISP site on the map server.
22. **ipv4 map-server**
23. **ipv6 map-server**
24. **ipv4 map-resolver**
25. **ipv6 map-resolver**
26. **exit**
27. **ip route** *prefix mask ip-address*
28. **ipv6 route** *ipv6-prefix/prefix-length ipv6-address*
29. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.2.1 255.255.255.252`	Configures an IPv4 address for the interface.

	Command or Action	Purpose
Step 5	**ipv6 address** *ipv6-address/ipv6-prefix* **Example:** `Device(config-if)# ipv6 address 2001:DB8:1::1/64`	Configures an IPv6 address for the interface.
Step 6	**router lisp** **Example:** `Device(config-if)# router lisp`	Enters LISP configuration mode.
Step 7	**locator-set** *locator-set-name* **Example:** `Device(config-router-lisp)# locator-set rtr-set1`	Specifies a locator set and enters LISP locator set configuration mode.
Step 8	*ipv4-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 10.0.3.1 priority 1 weight 1`	Configures the LISP locator set. The LISP locator set is the set of addresses the first-hop router uses when communicating with the gateway xTR. You can configure each locator address by creating a locator entry with assigned priority and weight.
Step 9	**exit** **Example:** `Device(config-router-lisp-locator-set)# exit`	Exits LISP locator set configuration mode and returns to LISP configuration mode.
Step 10	Repeat Step 7 to Step 9 to specify and configure another locator set.	—
Step 11	**locator-scope** *name* **Example:** `Device(config-router-lisp)# locator-scope s1`	Specifies the locator scope and enters locator scope configuration mode.
Step 12	**rtr-locator-set** *locator-set-name* **Example:** `Device(config-router-lisp-locator-scope)# rtr-locator-set rtr-set1`	Specifies the locator set of re-encapsulating tunnel router (RTR) to use in proxy reply for disjoint/cross address family routing locator (RLOC).
Step 13	**rloc-prefix** *ipv4-rloc-prefix* **Example:** `Device(config-router-lisp-locator-scope)# rloc-prefix 0.0.0.0/0`	Specifies the RLOC prefix to check against ingress tunnel router (ITR) RLOC and egress tunnel router (ETR) RLOC.
Step 14	**exit** **Example:** `Device(config-router-lisp-locator-set)# exit`	Exits LISP locator set configuration mode and returns to LISP configuration mode.
Step 15	Repeat Step 11 to Step 14 to specify and configure another locator scope.	—

	Command or Action	Purpose
Step 16	**site** *site-name* **Example:** `Device(config-router-lisp)# site R4`	Configures a LISP site on a map server and enters LISP site configuration mode.
Step 17	**authentication-key** *password* **Example:** `Device(config-router-lisp-site)# authentication-key R4KEY`	Specifies the authentication key that the LISP site uses.
Step 18	**eid-prefix** *ipv4-eid-prefix* **Example:** `Device(config-router-lisp-site)# eid-prefix 10.10.10.0/24`	Specifies a site IPv4 EID prefix.
Step 19	**eid-prefix** *ipv6-eid-prefix* **Example:** `Device(config-router-lisp-site)# eid-prefix 2001:DB8::/48`	Specifies a site IPv6 EID address prefix.
Step 20	**exit** **Example:** `Device(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 21	Repeat Step 16 to Step 20 to configure another LISP site on the map server.	—
Step 22	**ipv4 map-server** **Example:** `Device(config-router-lisp)# ipv4 map-server`	Enables IPv4 map server functionality.
Step 23	**ipv6 map-server** **Example:** `Device(config-router-lisp)# ipv6 map-server`	Enables IPv6 map server functionality.
Step 24	**ipv4 map-resolver** **Example:** `Device(config-router-lisp)# ipv4 map-resolver`	Enables IPv4 map resolver functionality.
Step 25	**ipv6 map-resolver** **Example:** `Device(config-router-lisp)# ipv6 map-resolver`	Enables IPv6 map resolver functionality.
Step 26	**exit** **Example:** `Device(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 27	**ip route** *prefix mask ip-address* **Example:** `Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.2.2`	Establishes static routes to the next hop destination.
Step 28	**ipv6 route** *ipv6-prefix/prefix-length ipv6-address* **Example:** `Device(config)# ipv6 route ::/0 2001:DB8:1::ABCD`	Establishes static routes to the next hop destination.
Step 29	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring RTR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ipv6 address** *ipv6-address/ipv6-prefix*
6. **router lisp**
7. **locator-set** *locator-set-name*
8. *ipv4-address* **priority** *priority-locator* **weight** *locator-weight*
9. *ipv6-address* **priority** *priority-locator* **weight** *locator-weight*
10. **exit**
11. **map-request itr-rlocs** *locator-set-name*
12. **eid-table default instance-id** *id*
13. **map-cache** *ipv4-EID-prefix* **map-request**
14. **map-cache** *ipv6-EID-prefix* **map-request**
15. **exit**
16. **ipv4 map-request-source** *source-address*
17. **ipv4 map-cache-limit** *cache-limit*
18. **ipv4 proxy-etr**
19. **ipv4 proxy-itr** *ipv4-local-locator ipv6-local-locator*
20. **ipv4 itr map-resolver** *map-resolver-address*
21. **ipv6 map-request-source** *source-address*
22. **ipv6 map-cache-limit** *cache-limit*
23. **ipv6 proxy-etr** *cache-limit*
24. **ipv6 proxy-itr** *ipv6-local-locator ipv4-local-locator*
25. **ipv6 itr map-resolver** *map-resolver-address*
26. **exit**

27. **ip route** *prefix mask ip-address*
28. **ipv6 route** *ipv6-prefix/prefix-length ipv6-address*
29. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.3.1 255.255.255.252`	Configures an IPv4 address for the interface.
Step 5	**ipv6 address** *ipv6-address/ipv6-prefix* **Example:** `Device(config-if)# ipv6 address 2001:DB8:2::1/64`	Configures an IPv6 address for the interface.
Step 6	**router lisp** **Example:** `Device(config-if)# router lisp`	Enters LISP configuration mode.
Step 7	**locator-set** *locator-set-name* **Example:** `Device(config-router-lisp)# locator-set setALL`	Specifies a locator set and enters LISP locator set configuration mode.
Step 8	*ipv4-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 10.0.3.1 priority 1 weight 1`	Configures an IPv4 or IPv6 address and policy for the re-encapsulation tunnel router (RTR).
Step 9	*ipv6-address* **priority** *priority-locator* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# 2001:DB8:2::1 priority 1 weight 1`	Configures an IPv4 or IPv6 address and policy for the RTR.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config-router-lisp-locator-set)# exit`	Exits LISP locator set configuration mode and returns to LISP configuration mode.
Step 11	**map-request itr-rlocs** *locator-set-name* **Example:** `Device(config-router-lisp)# map-request itr-rlocs setALL`	Configures the locator set to be used as routing locators (RLOCs) in the ingress tunnel router (ITR) RLOC field of Map-Request messages sent from the RTR.
Step 12	**eid-table default instance-id** *id* **Example:** `Device(config-router-lisp)# eid-table default instance-id 0`	Configures an association between the default (global) routing table and a LISP instance ID, and enters EID table configuration mode.
Step 13	**map-cache** *ipv4-EID-prefix* **map-request** **Example:** `Device(config-router-lisp-eid-table)# map-cache 0.0.0.0/0 map-request`	Configures static endpoint identifier-to-routing locator (EID-to-RLOC) mappings for an ITR and enables sending of Map-Request message for a LISP destination EID.
Step 14	**map-cache** *ipv6-EID-prefix* **map-request** **Example:** `Device(config-router-lisp-eid-table)# map-cache ::/0 map-request`	Configures static EID-to-RLOC mappings for an ITR and enables sending of Map-Request message for a LISP destination EID.
Step 15	**exit** **Example:** `Device(config-router-lisp-eid-table)# exit`	Exits LISP EID table configuration mode and returns to LISP configuration mode.
Step 16	**ipv4 map-request-source** *source-address* **Example:** `Device(config-router-lisp)# ipv4 map-request-source 10.0.3.1`	Specifies the IPv4 source address to be used in LISP IPv4 Map-Request messages. The ITR RLOCs configured under Steps 7 through 10, and Step 11 take precedence. However, this step (16) is still required.
Step 17	**ipv4 map-cache-limit** *cache-limit* **Example:** `Device(config-router-lisp)# ipv4 map-cache-limit 100000`	(Optional) Specifies maximum number of IPv4 LISP map cache entries allowed to be stored on the router. The valid range is from 0 to 100000.
Step 18	**ipv4 proxy-etr** **Example:** `Device(config-router-lisp)# ipv4 proxy-etr`	Configures a device to act as an IPv4 LISP proxy egress tunnel router (PETR).
Step 19	**ipv4 proxy-itr** *ipv4-local-locator ipv6-local-locator* **Example:** `Device(config-router-lisp)# ipv4 proxy-itr 10.0.3.1 2001:DB8:2::1`	Configures this device to act as an IPv4 proxy ingress tunnel router (PITR), and configures the IPv4 and IPv6 locator addresses used as a source address for encapsulation of data packets.

	Command or Action	Purpose

Step 20	**ipv4 itr map-resolver** *map-resolver-address*
	Example:
	`Device(config-router-lisp)# ipv4 itr map-resolver 10.0.2.1`
	`Device(config-router-lisp)# ipv4 itr map-resolver 2001:DB8:1::1`
		Configures a locator address for the LISP map resolver to which this device will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
		• The locator address of the map resolver may be an IPv4 or IPv6 address.
		Note You can configure up to 8 map resolvers if multiple map resolvers are available.

Step 21	**ipv6 map-request-source** *source-address* **Example:** `Device(config-router-lisp)# ipv6 map-request-source 2001:DB8:2::1`	The ITR RLOCs configured under Steps 7 through 10, and Step 11 take precedence. However, this step (16) is still required.
Step 22	**ipv6 map-cache-limit** *cache-limit* **Example:** `Device(config-router-lisp)# ipv6 map-cache-limit 100000`	(Optional) Specifies the maximum number of IPv6 LISP map cache entries allowed to be stored on the device. The valid range is from 0 to 100000.
Step 23	**ipv6 proxy-etr** *cache-limit* **Example:** `Device(config-router-lisp)# ipv6 proxy-etr`	Configures a device to act as an IPv6 LISP PETR.
Step 24	**ipv6 proxy-itr** *ipv6-local-locator ipv4-local-locator* **Example:** `Device(config-router-lisp)# ipv6 proxy-itr 2001:DB8:2::1 10.0.3.1`	Configures this device to act as an IPv6 PITR, and configures the IPv4 and IPv6 locator addresses used as a source address for encapsulation of data packets.

Step 25	**ipv6 itr map-resolver** *map-resolver-address*
	Example:
	`Device(config-router-lisp)#  ipv6 itr map-resolver 10.0.2.1`
	`Device(config-router-lisp)#  ipv6 itr map-resolver 2001:DB8:1::1`
		Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions.
		• The locator address of the map resolver may be an IPv4 or IPv6 address.
		Note You can configure up to eight map resolvers if multiple map resolvers are available.

Step 26	**exit** **Example:** `Device(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 27	**ip route** *prefix mask ip-address* **Example:** `Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.3.2`	Establishes static routes to the next hop destination.

	Command or Action	Purpose
Step 28	**ipv6 route** *ipv6-prefix/prefix-length ipv6-address* **Example:** `Device(config)# ipv6 route ::/0 2001:DB8:ABCD::1`	Establishes static routes to the next hop destination.
Step 29	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying LISP Support for Disjoint RLOC Domains

SUMMARY STEPS

1. **enable**
2. **show ip lisp database**
3. **show ipv6 lisp database**
4. **show lisp site detail**
5. **show ip lisp map-cache**
6. **show ipv6 lisp map-cache**

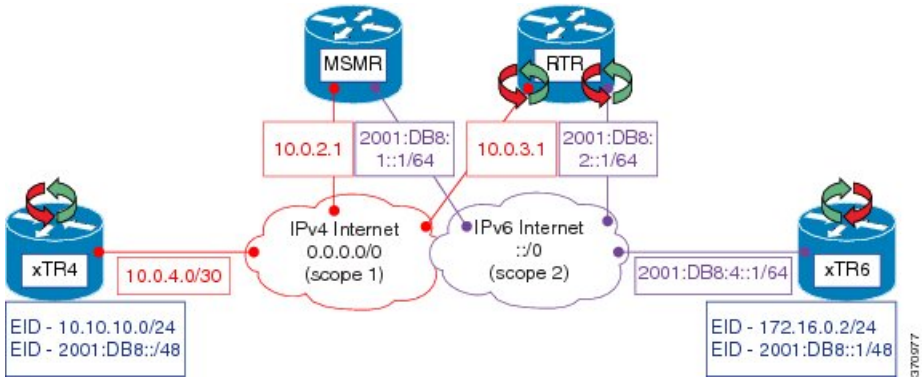
DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip lisp database** **Example:** `Device# show ip lisp database`	Displays Locator/ID Separation Protocol (LISP) egress tunnel router (ETR) configured local IPv4 endpoint identifier (EID) prefixes and associated locator sets.
Step 3	**show ipv6 lisp database** **Example:** `Device# show ipv6 lisp database`	Displays LISP ETR configured local IPv6 EID prefixes and associated locator sets.
Step 4	**show lisp site detail** **Example:** `Device# show lisp site detail`	Displays details of LISP sites configured on a LISP map server.
Step 5	**show ip lisp map-cache** **Example:** `Device# show ip lisp map-cache`	Displays the current dynamic and static IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) map cache entries.

	Command or Action	Purpose
Step 6	**show ipv6 lisp map-cache** **Example:** `Device# show ipv6 lisp map-cache`	Displays the current dynamic and static IPv6 EID-to-RLOC map cache entries.

Configuration Examples for LISP Support for Disjoint RLOC Domains

Figure 193: LISP - Disjoint RLOC Domains topology



The examples below show the complete configuration for the LISP topology illustrated in the figure above.

Example: Configuring xTR

The following example shows how to configure xTR4:

```
Device> enable
Device# configure terminal
Device(config)# interface loopback0
Device(config-if)# ip address 10.10.10.4 255.255.255.0
Device(config-if)# ipv6 address 2001:DB8:0:ABCD::1/64
Device(config-if)# interface ethernet0/0
Device(config-if)# ip address 10.0.4.1 255.255.255.252
Device(config-if)# router lisp
Device(config-router-lisp)# locator-set R4
Device(config-router-lisp-locator-set)# 10.0.4.1 priority 1 weight 1
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# eid-table default instance-id 0
Device(config-router-lisp-eid-table)# database-mapping 10.10.10.0/24 locator-set R4
Device(config-router-lisp-eid-table)# database-mapping  2001:DB8::/48 locator-set R4
Device(config-router-lisp-eid-table)# exit
Device(config-router-lisp)# ipv4 itr map-resolver 10.0.2.1
Device(config-router-lisp)# ipv4 itr
Device(config-router-lisp)# ipv4 etr map-server 10.0.2.1 key R4KEY
Device(config-router-lisp)# ipv4 etr
Device(config-router-lisp)# ipv6 itr map-resolver 10.0.2.1
Device(config-router-lisp)# ipv6 itr
```

```
Device(config-router-lisp)# ipv6 etr map-server 10.0.2.1 key R4KEY
Device(config-router-lisp)# ipv6 etr
Device(config-router-lisp)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.4.2
```

The following example shows how to configure xTR6:

```
Device> enable
Device# configure terminal
Device(config)# interface loopback0
Device(config-if)# ip address 172.16.0.4 255.255.255.0
Device(config-if)# ipv6 address 2001:DB8::4/64
Device(config-if)# interface ethernet0/0
Device(config-if)# ipv6 address 2001:DB8:4::2/64
Device(config-if)# router lisp
Device(config-router-lisp)# locator-set R6
Device(config-router-lisp-locator-set)# 2001:DB8:4::2 priority 1 weight 1
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# eid-table default instance-id 0
Device(config-router-lisp-eid-table)# database-mapping 172.16.0.2/24 locator-set R4
Device(config-router-lisp-eid-table)# database-mapping 2001:DB8::1/48 locator-set R4
Device(config-router-lisp-eid-table)# exit
Device(config-router-lisp)# ipv4 itr map-resolver 2001:DB8:3::2
Device(config-router-lisp)# ipv4 itr
Device(config-router-lisp)# ipv4 etr map-server 2001:DB8:3::2 key R4KEY
Device(config-router-lisp)# ipv4 etr
Device(config-router-lisp)# ipv6 itr map-resolver 2001:DB8:3::2
Device(config-router-lisp)# ipv6 itr
Device(config-router-lisp)# ipv6 etr map-server 2001:DB8:3::2 key R4KEY
Device(config-router-lisp)# ipv6 etr
Device(config-router-lisp)# exit
Device(config)# ipv6 route ::/0 2001:DB8:4::1
```

Example: Configuring MSMR

```
Device> enable
Device# configure terminal
Device(config)# interface ethernet0/0
Device(config-if)# ip address 10.0.2.1 255.255.255.252
Device(config-if)# ipv6 address 2001:DB8:1::1/64
Device (config-if)# router lisp
Device(config-router-lisp)# locator-set rtr-set1
Device(config-router-lisp-locator-set)# 10.0.3.1 priority 1 weight 1
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# locator-set rtr-set2
Device(config-router-lisp-locator-set)# 2001:DB8:2::1/64 priority 1 weight 1
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# locator-scope s1
Device(config-router-lisp-locator-scope)# rtr-locator-set rtr-set1
Device(config-router-lisp-locator-scope)# rloc-prefix 0.0.0.0/0
Device(config-router-lisp-locator-scope)# exit
Device(config-router-lisp)# locator-scope s2
Device(config-router-lisp-locator-scope)# rtr-locator-set rtr-set2
Device(config-router-lisp-locator-scope)# rloc-prefix ::/0
Device(config-router-lisp-locator-scope)# exit
Device(config-router-lisp)# site R4
Device(config-router-lisp-site)# authentication-key R4KEY
Device(config-router-lisp-site)# eid-prefix 10.10.10.0/24
Device(config-router-lisp-site)# eid-prefix 2001:DB8::/48
```

```
Device(config-router-lisp-site)# exit
Device(config-router-lisp)# site R6
Device(config-router-lisp-site)# authentication-key R6KEY
Device(config-router-lisp-site)# eid-prefix 172.16.0.2/24
Device(config-router-lisp-site)# eid-prefix 2001:DB8::1/48
Device(config-router-lisp-site)# exit
Device(config-router-lisp)# ipv4 map-server
Device(config-router-lisp)# ipv4 map-resolver
Device(config-router-lisp)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.2.2
Device(config)# ipv6 route ::/0 2001:DB8:1::ABCD
```

Example: Configuring RTR

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet0/0
Device(config-if)# ip address 10.0.3.1 255.255.255.252
Device(config-if)# ipv6 address 2001:DB8:2::1/64
Device (config-if)# router lisp
Device(config-router-lisp)# locator-set setALL
Device(config-router-lisp-locator-set)# 10.0.3.1 priority 1 weight 1
Device(config-router-lisp-locator-set)# 2001:DB8:2::1 priority 1 weight 1
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# map-request itr-rlocs setALL
Device(config-router-lisp)# eid-table default instance-id 0
Device(config-router-lisp-eid-table)# map-cache 0.0.0.0/0 map-request
Device(config-router-lisp-eid-table)# map-cache ::/0 map-request
Device(config-router-lisp-eid-table)# exit
Device(config-router-lisp)# ipv4 map-request-source 10.0.3.1
Device(config-router-lisp)# ipv4 map-cache-limit 100000
Device(config-router-lisp)# ipv4 proxy-etr
Device(config-router-lisp)# ipv4 proxy-itr 10.0.3.1 2001:DB8:2::1
Device(config-router-lisp)# ipv4 itr map-resolver 10.0.2.1
Device(config-router-lisp)# ipv4 itr map-resolver 2001:DB8:1::1
Device(config-router-lisp)# ipv6 map-request-source 2001:DB8:2::1
Device(config-router-lisp)# ipv6 map-cache-limit 100000
Device(config-router-lisp)# ipv6 proxy-etr
Device(config-router-lisp)# ipv6 proxy-itr 2001:DB8:2::1 10.0.3.1
Device(config-router-lisp)# ipv6 itr map-resolver 10.0.2.1
Device(config-router-lisp)# ipv6 itr map-resolver 2001:DB8:1::1
Device(config-router-lisp)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.3.2
Device(config)# ipv6 route ::/0 2001:DB8:ABCD::1
```

Example: Verifying LISP Support for Disjoint RLOC Domains

Sample Output for the show ip lisp database Command

To display Locator/ID Separation Protocol (LISP) egress tunnel router (ETR) configured local IPv4 endpoint identifier (EID) prefixes and associated locator sets, use the **show ip lisp database** command in privileged EXEC mode.

```
Device# show ip lisp database
.
.
.
```

```
10.10.10.0/24, locator-set R4
Locator Pri/Wgt Source State
10.0.4.1 1/1 cfg-addr site-self, reachable
```

Sample Output for the show ipv6 lisp database Command

To display LISP ETR configured local IPv6 EID prefixes and associated locator sets, use the **show ip lisp database** command in privileged EXEC mode.

```
Device# show ipv6 lisp database
.
.
.
2001:DB8::/48, locator-set R4
Locator Pri/Wgt Source State
10.0.4.1 1/1 cfg-addr site-self, reachable
 mm
```

Sample Output for the show lisp site detail Command

To display configured LISP sites on a LISP map server, use the **show lisp site detail** in privileged EXEC mode.

```
Device# show lisp site detail
.
.
.
Site name: R4
.
.
.
EID-prefix: 10.10.10.0/24
.
.
.
ETR 10.0.4.1, last registered 00:00:52, no proxy-reply, map-notify
TTL 1d00h, no merge, hash-function sha1, nonce 0x28517C31-0x7B233E66
state complete, no security-capability
xTR-ID 0xEC52ECC2-0x006CEAFE-0x814263B3-0x89675EB6
site-ID unspecified
Locator Local State Pri/Wgt Scope
10.0.4.1 yes up 1/1 s1
EID-prefix: 2001:DB8::/48
.
.
.
.
ETR 10.0.4.1, last registered 00:00:39, no proxy-reply, map-notify
TTL 1d00h, no merge, hash-function sha1, nonce 0xF91CB211-0x5B00E72C
state complete, no security-capability
xTR-ID 0xEC52ECC2-0x006CEAFE-0x814263B3-0x89675EB6
site-ID unspecified
Locator Local State Pri/Wgt Scope
10.0.4.1 yes up 1/1 s1
.
.
.
```

Sample Output for the show ip lisp map-cache Command

To display the current dynamic and static IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) map cache entries, use the **show ip lisp map-cache** command in privileged EXEC mode.

```
Device# show ip lisp map-cache

LISP IPv4 Mapping Cache for EID-table default (IID 0), 2 entries
.
.
.
172.16.0.2/24, uptime: 00:01:14, expires: 00:13:44, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.3.1  00:01:14  up           1/1
```

Sample Output for the show ipv6 lisp map-cache Command

To display the current dynamic and static IPv6 EID-to-RLOC map-cache entries, use the **show ipv6 lisp map-cache** command in privileged EXEC mode.

```
Device# show ipv6 lisp map-cache

LISP IPv6 Mapping Cache for EID-table default (IID 0), 2 entries
.
.
.
2001:DB8::1/48, uptime: 00:02:18, expires: 00:12:44, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.3.1  00:02:18  up           1/1
```

Additional References for LISP Support for Disjoint RLOC Domains

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Locator/ID Separation Protocol (LISP) commands	Cisco IOS IP Routing: LISP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Support for Disjoint RLOC Domains

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Release	Feature Information
LISP Support for Disjoint RLOC Domains		The LISP Support for Disjoint RLOC domains feature enables LISP-to-LISP communications between LISP sites that are connected to different RLOC spaces but have no connectivity to each other.

CHAPTER 186

LISP Data Plane Security

The Locator/ID Separation Protocol (LISP) Data Plane Security feature ensures that only traffic from within a LISP VPN can be decapsulated into the VPN. The feature is enforced when LISP packets are decapsulated by a tunnel router at the destination. Egress tunnel routers (ETRs) and proxy egress tunnel routers (PETRs) validate that the source Routing Locator (RLOC) address carried by the data packet is a member of the LISP VPN.

The solution relies on Unicast Reverse Path Forwarding (uRPF) being implemented in the RLOC network to ensure that the RLOC source addresses in LISP encapsulated data packets cannot be spoofed. Packets from outside the LISP VPN carry invalid source RLOCs that are blocked during decapsulation by ETRs and PETRs.

The advantages of implementing the LISP Data Plane Security feature are given below:

- Enhanced security due to validation by ETRs and PETRs during decapsulation.

Prerequisites for LISP Data Plane Security

- Understanding of LISP concepts, including the concept of virtual routing and forwarding (VRF) instances bound to instance IDs (IIDs). These concepts are explained in the chapters *LISP Overview*, *Configuring LISP*, and *LISP Shared Model Virtualization*.
- uRPF implementation in the RLOC network.

Restrictions for LISP Data Plane Security

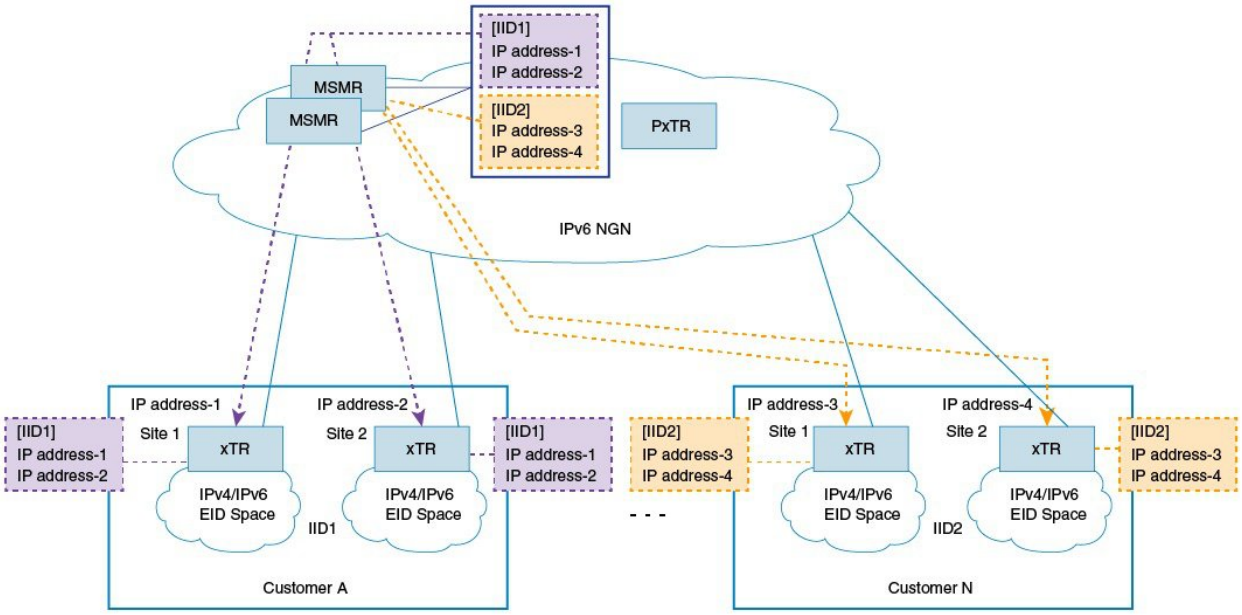
- All sites within a given LISP VPN must register to one or a common set of Map-Servers. That is, all IP prefixes associated with a specific instance ID must be delegated from common a Map-Server to ensure that these Map-Servers can construct a complete RLOC set for the given LISP VPN.

Information About LISP Data Plane Security

Source RLOC Decapsulation Filtering

This feature enhances data plane security by monitoring LISP packets during the decapsulation stage, when the packets are sent from an ingress tunnel router (ITR) or proxy ingress tunnel router (PITR) to an ETR or PETR. To protect LISP VPN end sites from decapsulating LISP packets that do not belong to the VPN, whether the result of misconfiguration or an attack, the source address in the incoming LISP packets are compared with a dynamically distributed set of source RLOC addresses corresponding to valid LISP VPN end sites. LISP packet decapsulation by ETRs and PETRs validate that a source RLOC address of an incoming LISP data packet is a member of the VPN. Note that this solution requires that source RLOC addresses are not spoofed, and hence unicast RPF or ingress anti-spoofing access control lists (ACLs) are required within the RLOC core network.

Consider the scenario in the image below:



1. Customer A has 2 LISP sites, site1 and site2, each having an xTR (a device performing the role of ETR and ITR). Site 1 and Site 2 register with the Map-Servers (of the Map-Server/Map-Resolver [MSMR] devices) supporting the LISP control plane for the LISP VPN with instance ID 1. The Map-Server automatically records the registration RLOCs for both sites, and dynamically pushes this list of valid RLOCs to both sites. In this way, site 1 and site 2 of the customer A LISP VPN can send traffic between each other. No other LISP encapsulated traffic is permitted, as the source RLOC will not match the valid source RLOC list.

2. Customer N also has 2 LISP sites, site1 and site2, and both register to the Map-Servers supporting the LISP control plane for this LISP VPN with instance ID 2. The Map-Server automatically records the registration RLOCs for both sites, and dynamically pushes this list of valid RLOCs to both sites. In this way, site 1 and site 2 of the customer N LISP VPN can send traffic between each other. No other LISP encapsulated traffic is permitted, as the source RLOC will not match the valid source RLOC list.

In addition to the automatically learned source RLOCs of registering LISP sites, the per-IID (instance ID) membership list can be extended to include specific source RLOCs of valid devices that do no register, such as PxTRs. When this feature is deployed, the source RLOCs of the PxTR is made available with the xTRs.

Some pointers for implementing source RLOC decapsulation filtering are given below :

- For Map-Servers to be able to construct the complete list of members for an EID instance ID, they must receive registrations from all the xTRs participating in the customer VPN.
- Map-Servers construct the EID instance ID-RLOC membership list using the RLOC information in the received mapping records in map-register messages.
- All IP prefixes associated with a specific instance ID must be delegated from a common Map-Server to ensure that these Map-Servers can construct a complete RLOC set for the given LISP VPN.
- All xTRs within a VPN must register with a common set of Map-Servers.
- PxTRs do not (normally) register with the Map-Servers, such that the Map-Servers could discover the PxTR RLOC, and that the Map-Servers could distribute learned RLOCs to the PxTRs. Thus, PxTR RLOCs need to be manually configured on the Map-Server.
- The EID instance membership lists built by Map-Servers are communicated only to xTRs and PxTRs that are members of the VPN.

TCP-based Sessions for LISP Packet Transport

The LISP data plane security mechanism requires the automated distribution and updating of RLOC filter lists to VPN members. This automated distribution is accomplished through a TCP-based session established between the xTRs and Map-Servers after the normal registration process has completed.

For example, xTRs periodically transmit map register messages and process the resulting map notify messages issued by the Map-Server. The Map-Servers process map register messages, update corresponding registration state, and transmit matching map notify messages.

To implement a more reliable, secure, and scalable transport option, TCP-based sessions are provided for LISP-related communication between xTRs and Map-Servers.

Some pointers regarding TCP-based sessions are given below:

- The UDP-based registration mechanism is conducted, and then a TCP-based session is established and used for the distribution of EID-instance RLOC membership lists. The number of xTRs that a Map-Server can support is limited by the number of TCP sessions that the Map-Server can establish and maintain. This determines the number of VPN customers that a Map-Server can host.
- The xTRs belonging to the same VPN must register with the same Map-Servers. This limits the number of sites within a VPN to the number of TCP sessions that a Map-Server can support.

How to Configure LISP Data Plane Security

Configuring MSMR

To configure the MSMR devices, perform the steps given below:

Note Steps 5 to 10 are optional. You can use those to modify the list of RLOC addresses (filter list) discovered by the Map-Server.

Before you begin

- Ensure that you have available any RLOCs associated with PxTRs serving within the LISP VPN.

SUMMARY STEPS

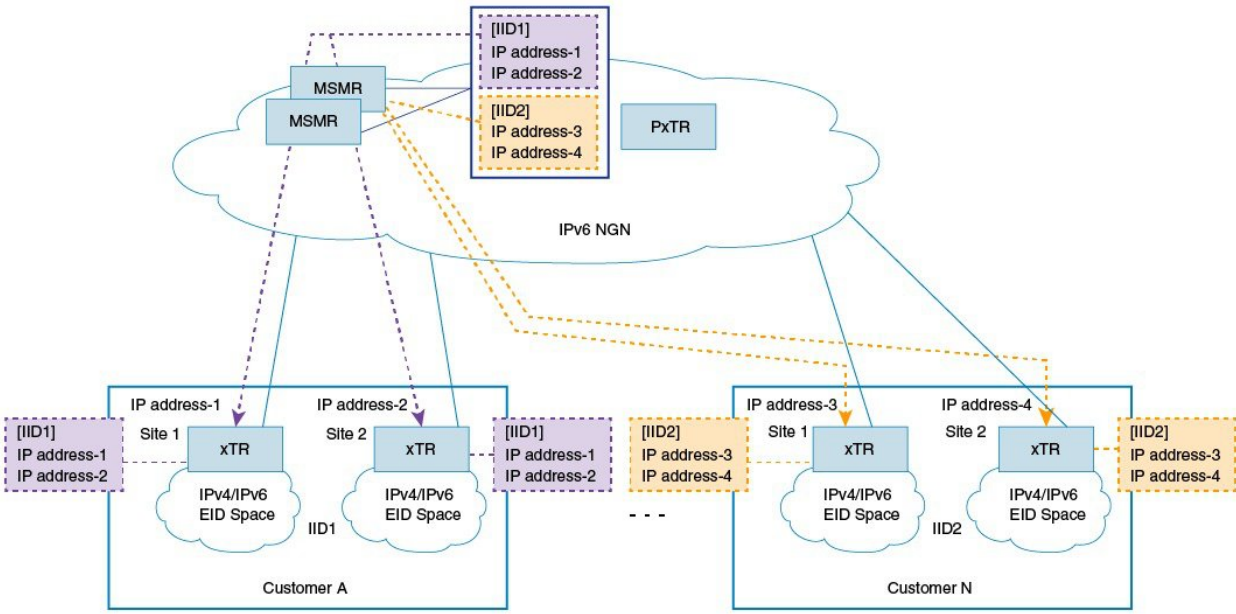
1. **enable**
2. **configure terminal**
3. **router lisp**
4. **map-server rloc members distribute**
5. **locator-set** *locator-set-name*
6. *ipv4-address* **priority** *value* **weight** *value*
7. **exit**
8. **eid-table vrf** *vrf-name* **instance-id** *iid*
9. **map-server rloc members modify-discovered add locator-set** *locator-set-name*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**map-server rloc members distribute** **Example:** `Device(config-router-lisp)# map-server rloc members distribute`	Enables distribution of the list of EID prefixes to xTRs at the customer end.
Step 5	**locator-set** *locator-set-name* **Example:** `Device(config-router-lisp)# locator-set PTR_set`	(Optional) Specifies a locator set for the PxTR and enters LISP locator set configuration mode.

	Command or Action	Purpose
Step 6	*ipv4-address* **priority** *value* **weight** *value* **Example:** `Device(config-router-lisp-locator-set)# 10.10.10.1 priority 1 weight 1`	(Optional) Configures the LISP locator set. You can configure each locator address by creating a locator entry with an assigned priority and weight
Step 7	**exit** **Example:** `Device(config-router-lisp-locator-set)# exit`	(Optional) Exits LISP locator set configuration mode and enters LISP configuration mode.
Step 8	**eid-table vrf** *vrf-name* **instance-id** *iid* **Example:** `Device(config-router-lisp)# eid-table vrf cust-A instance-id 1`	(Optional) Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode.
Step 9	**map-server rloc members modify-discovered add locator-set** *locator-set-name* **Example:** `Device(config-router-lisp-eid-table)# map-server rloc members modify-discovered add locator-set PTR_set`	(Optional) Adds RLOC addresses in the specified locator set to the list of *discovered* RLOC addresses. **Note** The updated list will be sent to the xTRs at the customer end when the distribution option is enabled.
Step 10	**exit** **Example:** `Device(config-router-lisp-eid-table)# exit`	(Optional) Exits eid-table configuration submode and enters LISP configuration mode.

Configuring the xTRs



To enable data plane security on the xTRs belonging to customer A (as shown in the image), configure the xTR at site1, as shown below:

Before you begin

- Ensure that you have configured the MSMR devices.
- Ensure that uRPF is implemented in the RLOC network.
- Ensure that you have identified EIDs and the LISP device acting as an xTR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **decapsulation filter rloc source member**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**decapsulation filter rloc source member** **Example:** `Device(config-router-lisp)# decapsulation filter rloc source member`	Enables source RLOC address validation of LISP packets.
Step 5	**exit** **Example:** `Device(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

What to do next

- The above steps enable data plane security for the xTR at one of customer A's sites, 'site1'. You need to repeat the steps to enable RLOC decapsulation filtering for customer A's second site, 'site2'.

Configuring PxTR

To configure the PxTR, perform the steps given below:

Before you begin

- Ensure that the MSMR devices and xTRs at the customer sites are configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **decapsulation filter rloc source members**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**decapsulation filter rloc source members** **Example:** `Device(config-router-lisp)# decapsulation filter rloc source members`	Enables source RLOC address validation of LISP packets.
Step 5	**exit** **Example:** `Device(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

What to do next

- Configure any other PxTR as needed.

Verifying LISP Data Plane Security On a Map-Server

Verify the LISP Data Plane Security feature on a Map-Server by using the commands given below:

SUMMARY STEPS

1. **show lisp** [**session** [**established**] | **vrf** [*vrf-name* [**session** [*peer-address*]]]]
2. **show lisp site rloc members** [**instance-id** *iid*]

DETAILED STEPS

Step 1 **show lisp** [**session** [**established**] | **vrf** [*vrf-name* [**session** [*peer-address*]]]]

Example:

```
Device# show lisp session

Sessions for VRF default, total: 8, established: 7
Peer                            State      Up/Down        In/Out    Users
2001:DB8:A:1::2                 Up         00:04:13        2/7      2
2001:DB8:A:2::2                 Up         00:04:13        2/7      2
2001:DB8:A:3::2                 Up         00:03:53        2/7      2
2001:DB8:B:1::2                 Up         00:04:04        2/6      2
2001:DB8:B:2::2                 Init       never           0/0      1
2001:DB8:C:1::2                 Up         00:03:55        2/6      2
2001:DB8:C:2::2                 Up         00:03:54        2/6      2
2001:DB8:E:F::2                 Up         00:04:04        6/19     4
```

This command displays reliable transport session information. If there is more than one transport session, the corresponding information will be displayed.

Step 2 **show lisp site rloc members** [**instance-id** *iid*]

Example:

```
Device# show lisp site rloc members

LISP RLOC membership for EID table default (IID 0), 5 entries

RLOC                        Origin                      Valid
10.0.1.2                    registration                 Yes
10.0.2.2                    config & registration        Yes
```

The **Origin** column displays configuration details of the RLOC member – whether the RLOC member is manually configured, automatically gleaned from received registrations, or both. The **Valid** column shows whether the RLOC is a valid member that is distributed to (P)xTRs. A listed RLOC may not be valid if it is gleaned from registrations but the 'override' option is used in the 'modify-discovered' configuration, and the specified locator-set does not include the RLOC.

Verifying and Troubleshooting LISP Data Plane Security on an xTR or PxTR

Verify the LISP Data Plane Security feature on an xTR or PxTR by using the commands given below:

SUMMARY STEPS

1. **show lisp** [**session** [**established**] | **vrf** [*vrf-name* [**session** [*peer-address*]]]]
2. **show lisp decapsulation filter** [*IPv4-rloc-address* | *IPv6-rloc-address*] [**eid-table** *eid-table-vrf* | **instance-id** *iid*]
3. **show cef source-filter table**

4. **debug lisp control-plane eid-membership**
5. **debug lisp control-plane session**

DETAILED STEPS

Step 1 **show lisp** [**session** [**established**] | **vrf** [*vrf-name* [**session** [*peer-address*]]]]

Example:

```
Device# show lisp session

Sessions for VRF default, total: 8, established: 7
Peer                            State      Up/Down        In/Out    Users
2001:DB8:A:1::2                 Up         00:04:13        2/7      2
2001:DB8:A:2::2                 Up         00:04:13        2/7      2
2001:DB8:A:3::2                 Up         00:03:53        2/7      2
2001:DB8:B:1::2                 Up         00:04:04        2/6      2
2001:DB8:B:2::2                 Init       never           0/0      1
2001:DB8:C:1::2                 Up         00:03:55        2/6      2
2001:DB8:C:2::2                 Up         00:03:54        2/6      2
2001:DB8:E:F::2                 Up         00:04:04        6/19     4
```

This command displays reliable transport session information. If there is more than one transport session, the corresponding information will be displayed.

Step 2 **show lisp decapsulation filter** [*IPv4-rloc-address* | *IPv6-rloc-address*] [**eid-table** *eid-table-vrf* | **instance-id** *iid*]

Example:

```
Device# show lisp decapsulation filter instance-id 0

LISP decapsulation filter for EID-table default (IID 0), 3 entries

Source RLOC     Added by
10.0.0.1         Config
10.0.0.5         209.165.200.230 209.165.200.232
10.0.0.6         Config 209.165.200.230
```

The RLOC address configuration details (whether it is manually configured or discovered) on a (P)xTR is displayed in the above table.

Step 3 **show cef source-filter table**

Example:

```
Device# show cef source-filter table

[lisp:0:0:IPv4] state [enabled, active], 0 entries, refcount 3, flags [], action [drop]
Database epoch 0
Hits 0, misses 0, fwd 0, drop 0
```

This command displays Cisco Express Forwarding (CEF) source-filter tables.

Step 4 **debug lisp control-plane eid-membership**

Example:

```
Device# debug lisp control-plane eid-membership
```

```
LISP control plane EID membership debugging is on
```

Displays debugging information for EID membership discovery.

Step 5 **debug lisp control-plane session**

Example:

```
Device# debug lisp control-plane session

LISP control plane session debugging is on
```

Displays detailed session establishment debugging information.

Configuration Examples for LISP Data Plane Security

Example: Configuring MSMR

Note Steps for adding the locator set and the RLOC address are optional. You can use those steps to modify the list of RLOC addresses (filter list) discovered by the Map-Server.

```
Device> enable
Device# configure terminal
Device(config)# router lisp
Device(config-router-lisp)# map-server rloc members distribute
Device(config-router-lisp)# locator-set PTR_set
Device(config-router-lisp-locator-set)# 10.10.10.1 priority 1 weight 1
Device(config-router-lisp-locator-set)# exit
Device(config-router-lisp)# eid-table vrf cust-A instance-id 1
Device(config-router-lisp-eid-table)# map-server rloc members modify-discovered add
locator-set PTR_set
Device(config-router-lisp-eid-table)# exit
```

Repeat the above steps to configure one or more map servers, as needed

Example: Configuring the xTRs

```
Device> enable
Device# configure terminal
Device(config)# router lisp
Device(config-router-lisp)# decapsulation filter rloc source member
Device(config-router-lisp)# exit
```

The above steps enable data plane security for the xTR at one of customer sites. You must repeat the steps to enable RLOC decapsulation filtering for other sites.

Example: Configuring PxTR

```
Device> enable
Device# configure terminal
Device(config)# router lisp
Device(config-router-lisp)# decapsulation filter rloc source member
Device(config-router-lisp)# exit
```

Additional References for LISP Data Plane Security

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Locator/ID Separation Protocol (LISP) commands	Cisco IOS IP Routing: LISP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 6830	*Locator/ID Separation Protocol (LISP)*
RFC 6832	*Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites*
RFC 6833	*Locator/ID Separation Protocol (LISP) Map-Server Interface*

MIBs

MIB	MIBs Link
• CISCO-LISP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Data Plane Security

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 196: Feature Information for LISP Data Plane Security

Feature Name	Releases	Feature Information
LISP Data Plane Security		The LISP Data Plane Security feature ensures that only traffic from within a LISP VPN can be decapsulated into the VPN. The following commands were introduced by this feature: **clear lisp vrf**, **decapsulation filter rloc source**, **debug lisp control-plane eid-membership**, **debug lisp control-plane session**, **map-server rloc members distribute**, **map-server rloc members modify-discovered**, **show lisp decapsulation filter**, **show lisp site rloc members**, **show lisp session** .

CHAPTER 187

LISP Reliable Registration

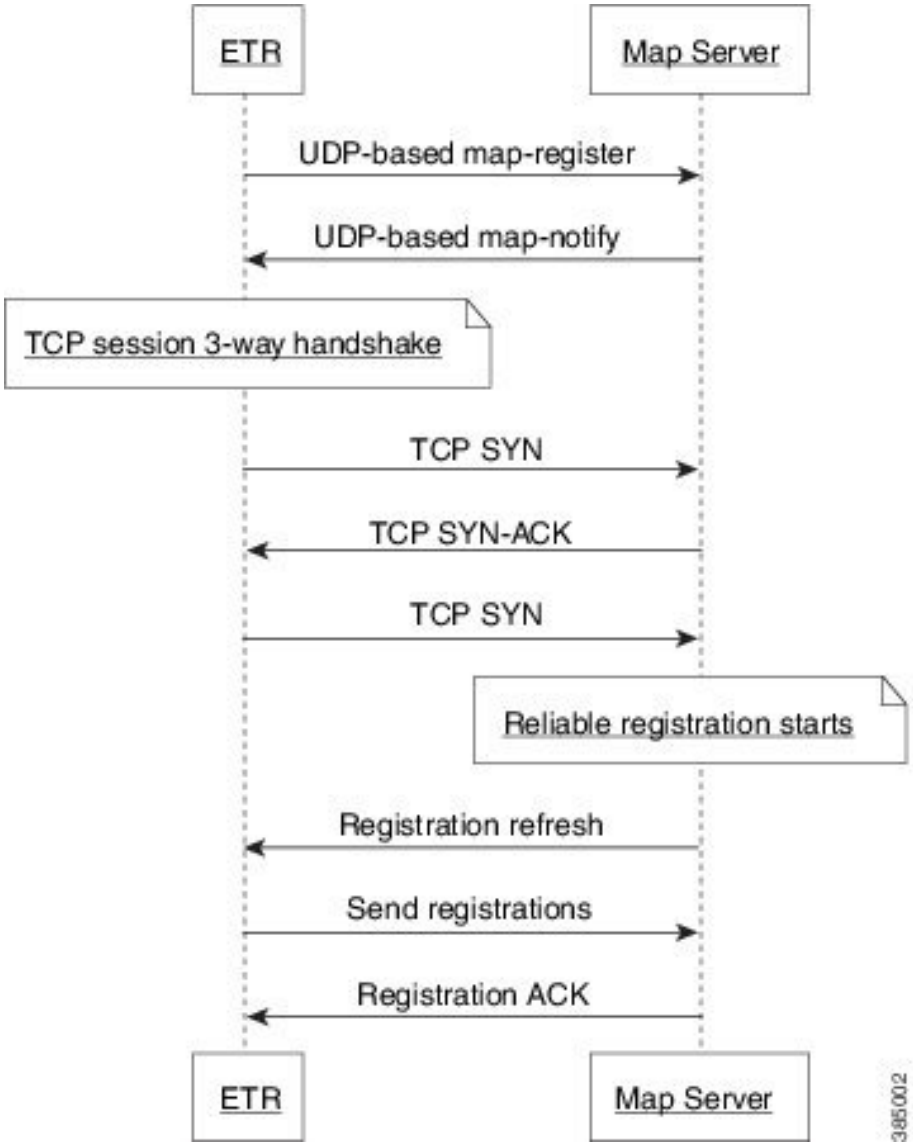
The LISP Reliable Registration feature supports establishment of TCP based reliable map registration between Egress Tunnel Router (ETR) and Map Server (MS).

Information About LISP Reliable Registration

LISP Reliable Map Registration

LISP ETR periodically sends UDP based map registration message to map server. This results in control traffic and scalability problems. TCP based reliable map registration or LISP reliable map registration mechanism is developed as an enhancement and replacement to the UDP based map registration mechanism.

Figure 194: LISP Reliable Map Registration Mechanism



The LISP reliable map registration mechanism as shown in the figure is described below:

- ETR sends UDP based map registration message to map server.
- Map server processes map registration and sends map-notify to ETR. This message serves as acknowledgment.
- ETR initiates a TCP session with map-server using three-way handshake.

Note When TCP based map registration is not supported by map server then ETR uses UDP based map registration to establish a session with the map server.

- Once the TCP session is established, map-server sends a registration refresh message to the ETR.

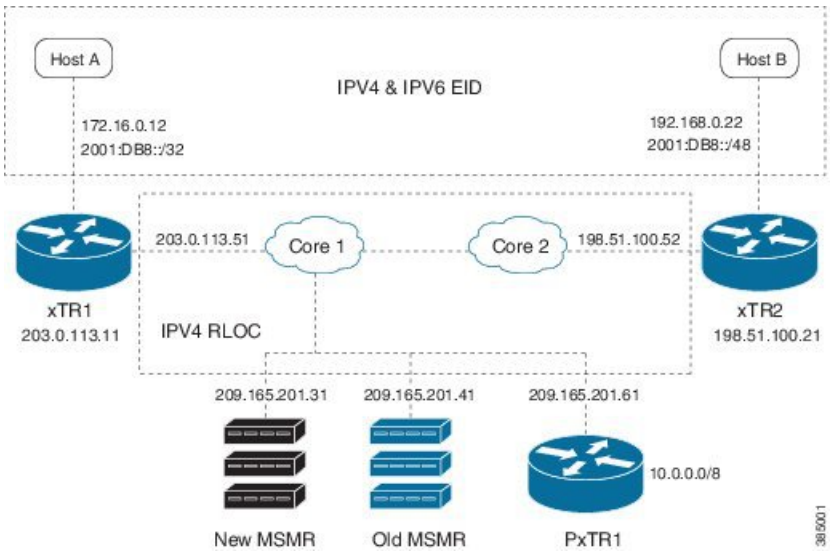
- ETR sends map registrations to the map server through the TCP connection.
- Map server acknowledges for the map registrations.

Note There are no configuration commands for this feature. This feature is turned on automatically.

Verifying the LISP Reliable Registration

Perform this task to verify the LISP Reliable Registration feature which is enabled automatically in the LISP network. In this example, a LISP site uses a single edge router that functions as both ITR and ETR (known as an xTR). Routing Locators (RLOCs) are in IPv4. EID prefixes are in both IPv4 and IPv6. The LISP site registers to two map server/map resolver (MSMR) devices in the network core. The topology used in verifying LISP Reliable Registration is as shown in the figure below.

Figure 195: LISP Reliable Registration Topology



The components as shown in the topology are described below:

- xTR1 and xTR2 are xTRs for 2 LISP sites.
- Core1 and Core 2 are routing locators (RLOCs) core routers with no LISP configuration.
- New MSMR is a map-server and map-resolver with reliable map-registration support, whereas Old MSMR does not support reliable map-registration.
- PxTR1 works as a Proxy Ingress Tunnel Router (PITR) and Proxy Egress Tunnel Router (PETR) between the network with 10.0.0.0/8 prefix and the LISP sites.
- Only static routing protocols are used in this setup to reduce control traffic.

In the following output, a '#' sign in the 'Up' column indicates reliable map registration session.

```
Device# show lisp site

LISP Site Registration Information
* = Some locators are down or unreachable
# = Some registrations are sourced by reliable transport
```

```
Site Name      Last         Up        Who Last         Inst       EID Prefix
               Register                Registered       ID
A              never        no            --                     0.0.0.0/0
               01:59:44     yes#      203.0.113.11               10.10.10.0/24
               01:59:44     yes#      203.0.113.11               10.20.20.0/24
               01:59:44     yes#      203.0.113.11               172.16.0.0/24
               01:59:44     yes#      203.0.113.11               2001:DB8::/32
B              never        no            --                     0.0.0.0/0
               never        no            --                     10.0.0.0/8
               01:59:43     yes#      198.51.100.21              10.30.30.0/24
               01:59:43     yes#      198.51.100.21              10.40.40.0/24
               never        no            --                     21.0.0.0/8
               01:59:43     yes#      198.51.100.21              21.21.21.0/24
               01:59:43     yes#      198.51.100.21              2001:DB8::/48
```

In the following output, no '#' sign in the 'Up' column indicates that the Old MSMR does not support reliable map registration.

```
Device# show lisp site

LISP Site Registration Information
* = Some locators are down or unreachable


Site Name      Last         Up        Who Last         Inst       EID Prefix
               Register                Registered       ID
A              never        no            --                     0.0.0.0/0
               00:00:00     yes       203.0.113.11               172.16.0.0/24
               00:00:55     yes       198.51.100.21              21.21.21.0/24
               00:00:03     yes       203.0.113.11               2001:DB8::/32
B              never        no            --                     10.0.0.0/8
               00:00:00     yes       203.0.113.11               10.10.10.0/24
               00:00:00     yes       203.0.113.11               10.20.20.0/24
               00:00:55     yes       198.51.100.21              10.30.30.0/24
               00:00:55     yes       198.51.100.21              10.40.40.0/24
               00:00:52     yes       198.51.100.21              2001:DB8::/48
```

The following output is from xTR1 that uses 2 map servers. Reliable map-registration session is established with 209.165.201.31 (New MSMR), but not with 209.165.201.41 (Old MSMR).

```
Device# show lisp session

Sessions for VRF default, total: 2, established: 1
Peer                          State     Up/Down       In/Out   Users
209.165.201.31                Up         05:05:40       6/3     2
209.165.201.41                Down       never          0/0     1
```

The following output is from New MSMR. It has established reliable map-registration sessions with two ETRs.

```
Device# show lisp session

Sessions for VRF default, total: 2, established: 2
Peer                          State     Up/Down       In/Out   Users
203.0.113.11                  Up         05:19:53       3/6     1
198.51.100.21                 Up         05:18:28       2/5     1
```

Additional References for LISP Reliable Registration

Related Documents

Document Title	Location
Cisco IOS commands	Cisco IOS Master Command List, All Releases
LISP commands	Cisco IOS IP Routing: LISP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 6830	*The Locator/ID Separation Protocol (LISP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Reliable Registration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 197: Feature Information for LISP Reliable Registration

Feature Name	Releases	Feature Information
LISP Reliable Registration		The LISP Reliable Registration feature supports establishment of TCP based reliable map-registration between Egress Tunnel Router (ETR) and Map Server (MS). The following commands were modified: **show lisp site**.

CHAPTER 188

Overlapping Prefix

The Overlapping prefix feature supports Endpoint Identifier (EID) registration by two sites where the EID prefix from one LISP site is a subset of the EID prefix from another LISP site.

Prerequisites for Overlapping Prefix

- Reliable registration must be established between the xTR (performs functions of both Egress Tunnel Router and Ingress Tunnel Router components) and map server/map resolver (MS/MR).

Information About Overlapping Prefix

Endpoint ID (EID)

An EID value for IPv4 is 32 bit and EID value for IPv6 is 128-bit. EIDs are used in the source and destination address fields of the first LISP header of a packet.

EID-Prefix

An EID-Prefix is a power-of-two blocks of EIDs allocated to a LISP site by an address allocation authority.

Map Server/Map Resolver (MS/MR)

MS and MR functions are implemented on the same device, which is referred to as an MS/MR device.

How to Configure Overlapping Prefix

Configuring Overlapping Prefix

Configure EID-prefix with "accept-more-specifies" keyword to allow MS to accept registration of more specific prefix.

```
router lisp
 site site3
  authentication-key cisco
  eid-prefix 172.16.0.0/8 accept-more-specifics
  exit
```

Register 3.0.0.0/8 with MS.

```
router lisp
 database-mapping 172.16.0.0/8 10.0.0.3 priority 1 weight 100
```

Register 3.1.0.0/16 with MS, which is more specific and overlap with 3.0.0.0/8 prefix registered from xTR3.

```
router lisp
 database-mapping 192.168.0.0/16 10.0.0.4 priority 1 weight 100
 database-mapping 192.0.2.0/8 10.0.0.4 priority 1 weight 100
```

Verifying Overlapping Prefix

Perform this task to verify the Overlapping Prefix feature in the LISP network. In this example, there are four routers: MSMR, xTR2, xTR3, and xTR4. Each router has an interface connection in the same subnet (RLOC space) 10.0.0.0/24. The following are the IP addresses of the routers:

Router	IP Address
MSMR	10.0.0.1
xTR2	10.0.0.2
xTR3	10.0.0.3
xTR4	10.0.0.4

MS/MR Output:

```
Device# show lisp site

LISP Site Registration Information
* = Some locators are down or unreachable
# = Some registrations are sourced by reliable transport

Site Name      Last      Up     Who Last             Inst     EID Prefix
               Register         Registered           ID
site2          00:15:08  yes#   10.0.0.2                      2.0.0.0/8
site3          00:15:05  yes#   10.0.0.3                      3.0.0.0/8
               00:15:01  yes#   10.0.0.4                      3.1.0.0/16
site4          00:15:01  yes#   10.0.0.4                      4.0.0.0/8
```

xTR1 Output:

```
Device# show ip lisp map-cache

LISP IPv4 Mapping Cache for EID-table default (IID 0), 3 entries

0.0.0.0/0,  uptime: 00:18:05, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
3.0.0.0/8,  uptime: 00:00:16, expires: 23:59:43, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.3 00:00:16  up           1/100
3.1.0.0/16, uptime: 00:00:08, expires: 23:59:51, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.4  00:00:08  up           1/100
```

xTR2 Output:

```
Device# show ip lisp map-cache

LISP IPv4 Mapping Cache for EID-table default (IID 0), 3 entries

0.0.0.0/0, uptime: 00:18:44, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2.0.0.0/8, uptime: 00:00:57, expires: 23:59:02, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.2  00:00:57  up           1/100
3.1.0.0/16, uptime: 00:18:40, expires: 23:42:12, via map-reply, self, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.4  00:17:47  up           1/100

Device# show ip lisp away

LISP Away Table for router lisp 0 (default) IID 0
Entries: 1

Prefix                                  Producer
3.1.0.0/16                              mapping-notification
```

xTR3 Output:

```
Device# show ip lisp map-cache

LISP IPv4 Mapping Cache for EID-table default (IID 0), 2 entries

0.0.0.0/0, uptime: 00:19:26, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2.0.0.0/8, uptime: 00:01:35, expires: 23:58:24, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.2  00:01:35  up           1/100

Device# show ip lisp away

LISP Away Table for router lisp 0 (default) IID 0
Entries: 0
```

Additional References for Overlapping Prefix

Related Documents

Document Title	Location
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Document Title	Location
LISP commands	Cisco IOS IP Routing: LISP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 6830	*The Locator/ID Separation Protocol (LISP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Overlapping Prefix

Table 198: Feature Information for Overlapping Prefix

Feature Name	Releases	Feature Information
Overlapping Prefix		The Overlapping prefix feature supports Endpoint Identifier (EID) registration by two sites where the EID prefix from one LISP site is a subset of the EID prefix from another LISP site. The following commands were modified: **authentication-key**, **database-mapping**, **router lisp**.

CHAPTER 189

LISP Generalized SMR

The LISP Generalized SMR feature enables LISP xTR (ITR and ETR) to update map cache when there is a change in database mapping.

Note There is no configuration commands for this feature. This feature is turned on automatically.

Information About LISP Generalized SMR

Solicit-Map-Request (SMR)

Soliciting a Map-Request enables ETRs to control requests for Map-Reply messages when there is change in database mapping. SMRs enable remote ITRs to update the database mappings that are cached. An SMR message is simply a bit set in a Map-Request message. An ITR or PITR will send a Map-Request when they receive an SMR message.

Note There is no configuration commands for this feature. This feature is turned on automatically.

Generalized SMR (GSMR)

SMR was mainly used to support LISP mobility. This mechanism has been generalized (Generalized Solicit Map Request - GSMR) to support the following use cases:

- De-configured local EID
- Local EID no-route (when an ETR decapsulates a data packet and finds no route for a configured local EID)

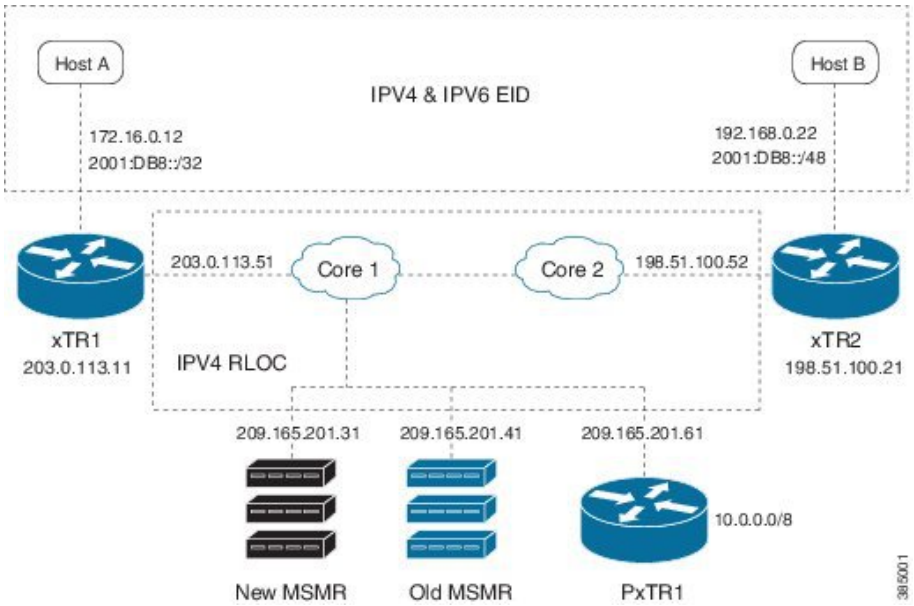
- Mobility host move out and detection
- Overlapping prefix

Note There are no configuration commands for this feature. This feature is turned on automatically.

Verifying LISP Generalized SMR

Perform this task to verify the LISP Generalized SMR feature which is enabled automatically in the LISP network. In this example, a LISP site uses a single edge router that functions as both ITR and ETR (known as an xTR). Routing Locators (RLOCs) are in IPv4. EID prefixes are in both IPv4 and IPv6. The LISP site registers to two map server/map resolver (MSMR) devices in the network core. The topology used in verifying LISP Generalized SMR is as shown in the figure below.

Figure 196: LISP Generalized SMR Topology



The components as shown in the topology are described below:

- xTR1 and xTR2 are xTRs for 2 LISP sites.
- Core1 and Core 2 are routing locators (RLOCs) core routers with no LISP configuration.
- New MSMR is a map-server and map-resolver with reliable map-registration support, whereas Old MSMR does not support reliable map-registration.
- PxTR1 works as a Proxy Ingress Tunnel Router (PITR) and Proxy Egress Tunnel Router (PETR) between the network with 10.0.0.0/8 prefix and the LISP sites.
- Only static routing protocols are used in this setup to reduce control traffic.

Verifying 172.16.0.0/24 is in map cache on xTR2:

```
Device# show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 3 entries
```

```
0.0.0.0/0, uptime: 03:32:45, expires: never, via static send map-request
Negative cache entry, action: send-map-request
10.20.20.0/24, uptime: 00:00:05, expires: 23:59:54, via map-reply, complete
Locator       Uptime    State     Pri/Wgt
203.0.113.11   00:00:05 up           1/100
172.16.0.0/24, uptime: 00:35:49, expires: 23:24:10, via map-reply, complete
Locator       Uptime   State     Pri/Wgt
203.0.113.11   00:35:49 up           1/100
```

Shutting down interface Ethernet1/0 on xTR1:

```
Device(config)# interface ethernet 1/0
Device(config-if)# shutdown
```

Verifying 172.16.0.0/24 is in map cache on xTR1:

```
Device# show ip lisp data
LISP ETR IPv4 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 3, no-route 1, inactive 0
10.10.10.0/24, locator-set set1
Locator       Pri/Wgt Source     State
203.0.113.11   1/100   cfg-addr   site-self, reachable
10.20.20.0/24, locator-set set1
Locator       Pri/Wgt Source     State
203.0.113.11   1/100   cfg-addr   site-self, reachable
172.16.0.0/24, locator-set set1 *** NO ROUTE TO EID PREFIX ***
Locator       Pri/Wgt Source     State
203.0.113.11   1/100   cfg-addr   site-self, reachable
```

Pinging Host A from Host B:

```
Device# ping 172.16.0.12
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.12, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

xTR1 decapsulates the data packets, finds out the no-route situation, and sends an SMR to xTR2:

```
Device#
*Feb 19 22:08:15.160: LISP: Send map request type dyn-EID SMR
*Feb 19 22:08:15.160: LISP: Send map request for EID prefix IID 0 192.168.0.22/32
*Feb 19 22:08:15.160: LISP-0: AF IID 0 IPv4, Send SMR map-request for 172.16.0.12 to
198.51.100.21.
*Feb 19 22:08:15.160: LISP-0: EID-AF IPv4, Sending probe map-request from 203.0.113.11 to
198.51.100.21
for EID 21.21.21.22/32, ITR-RLOCs 1, nonce 0x68E45971-0xE3DF4931, SMR 172.16.0.12, DoNotReply.
```

xTR2 processes the SMR and sends out a map-request to the map server:

```
Device#
*Feb 19 22:08:15.161: LISP: Processing received Map-Request(1) message on Ethernet0/0 from
 203.0.113.11:4342 to 198.51.100.21:4342
*Feb 19 22:08:15.161: LISP: Received map request for IID 0 192.168.0.22/32, source_eid IID
 0 172.16.0.12, ITR-RLOCs: 203.0.113.11,
records 1, nonce 0x68E45971-0xE3DF4931, probe, SMR, DoNotReply
*Feb 19 22:08:15.161: LISP-0: AF IID 0 IPv4, Scheduling SMR trigger Map-Request for
172.16.0.12/32 from 192.168.0.22.
*Feb 19 22:08:15.161: LISP-0: IID 0 SMR & D bit set, not replying to map-request.
*Feb 19 22:08:15.290: LISP: Send map request type SMR
*Feb 19 22:08:15.290: LISP: Send map request for EID prefix IID 0 172.16.0.12/32
Device#
*Feb 19 22:08:15.290: LISP-0: AF IID 0 IPv4, Send SMR triggered map request for 172.16.0.12/32
 (1) from 192.168.0.22.
*Feb 19 22:08:15.290: LISP-0: EID-AF IPv4, Sending map-request from 172.16.0.12 to 172.16.0.12
 for EID 172.16.0.12/32, ITR-RLOCs 1,
nonce 0x4D04AB2F-0x99FF6FF5 (encap src 198.51.100.21, dst 209.165.201.41).
```

```
Device#
*Feb 19 22:08:16.333: LISP: Send map request type SMR
*Feb 19 22:08:16.333: LISP: Send map request for EID prefix IID 0 172.16.0.12/32
*Feb 19 22:08:16.333: LISP-0: AF IID 0 IPv4, Send SMR triggered map request for 172.16.0.12/32
 (2) from 192.168.0.22.
*Feb 19 22:08:16.333: LISP-0: EID-AF IPv4, Sending map-request from 172.16.0.12 to 172.16.0.12
 for EID 172.16.0.12/32, ITR-RLOCs 1,
nonce 0x4D04AB2F-0x99FF6FF5 (encap src 198.51.100.21, dst 209.165.201.41).
Device#
*Feb 19 22:08:18.423: LISP-0: Map Request IID 0 prefix 172.16.0.12/32 SMR[LL], Switching
Map-Resolver 209.165.201.41 to 209.165.201.31.
*Feb 19 22:08:18.423: LISP: Send map request type SMR
*Feb 19 22:08:18.423: LISP: Send map request for EID prefix IID 0 172.16.0.12/32
*Feb 19 22:08:18.423: LISP-0: AF IID 0 IPv4, Send SMR triggered map request for 172.16.0.12/32
 (3) from 192.168.0.22.
*Feb 19 22:08:18.423: LISP-0: EID-AF IPv4, Sending map-request from 172.16.0.12 to 172.16.0.12
 for EID 172.16.0.12/32, ITR-RLOCs 1,
nonce 0x5A4AC708-0x59A42AB6 (encap src 198.51.100.21, dst 209.165.201.31).
*Feb 19 22:08:18.424: LISP: Processing received Map-Reply(2) message on Ethernet0/0 from
209.165.201.31:4342 to 198.51.100.21:4342
*Feb 19 22:08:18.424: LISP: Received map reply nonce 0x5A4AC708-0x59A42AB6, records 1
```

xTR2's map-cache is updated upon map-reply from the map server:

```
Device# show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 3 entries

0.0.0.0/0, uptime: 03:56:43, expires: never, via static send map-request
Negative cache entry, action: send-map-request
10.20.20.0/24, uptime: 00:24:04, expires: 23:35:56, via map-reply, complete
Locator       Uptime   State     Pri/Wgt
203.0.113.11   00:24:04 up          1/100
172.16.0.10/24, uptime: 00:59:48, expires: 00:00:51, via map-reply, forward-native
Negative cache entry, action: forward-native
```

xTR1 will put the 172.16.0.10/24 prefix in its away table:

```
Device# show ip lisp away
LISP Away Table for router lisp 0 (default) IID 0
Entries: 1
Prefix                                    Producer
172.16.0.10/24                            local EID
```

Additional References for LISP Reliable Registration

Related Documents

Document Title	Location
Cisco IOS commands	Cisco IOS Master Command List, All Releases
LISP commands	Cisco IOS IP Routing: LISP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 6830	*The Locator/ID Separation Protocol (LISP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Generalized SMR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 199: Feature Information for LISP Generalized SMR

Feature Name	Releases	Feature Information
LISP Generalized SMR		The LISP Generalize SMR feature supports LISP mobility, de-configured local Endpoint Identifier (EID), local EID no-route, overlapping prefix support, and mobility host move out and detection. The following commands were modified: **show ip lisp away**, **show ip lisp data**, **show ip lisp map-cache**.

CHAPTER 190

TTL Propagate Disable and Site-ID Qualification

The TTL Propagate Disable feature supports disabling of the TTL (Time-To-Live) propagation for implementing the traceroute tool in a LISP network when RLOC and EID belong to different address-family.

The Site ID Qualification feature supports Endpoint Identifier (EID) prefix registration by multiple LISP sites.

Information About TTL Propagate Disable and Site-ID Qualification

LISP Site

LISP site is a set of routers in an edge network that are under a single technical administration. LISP routers in the edge network are the demarcation points to separate the edge network from the core network.

Map Server (MS)

An MS implements part of the distributed LISP mapping database by accepting registration requests from its client Egress Tunnel Routers (ETRs) and aggregating the successfully registered EID prefixes of ETRs.

Routing Locator (RLOC)

An RLOC is an IPv4 or IPv6 address of an Egress Tunnel Router (ETR).

Traceroute Tool

The traceroute tool is used to discover the routes that packets take when traveling to their destination.

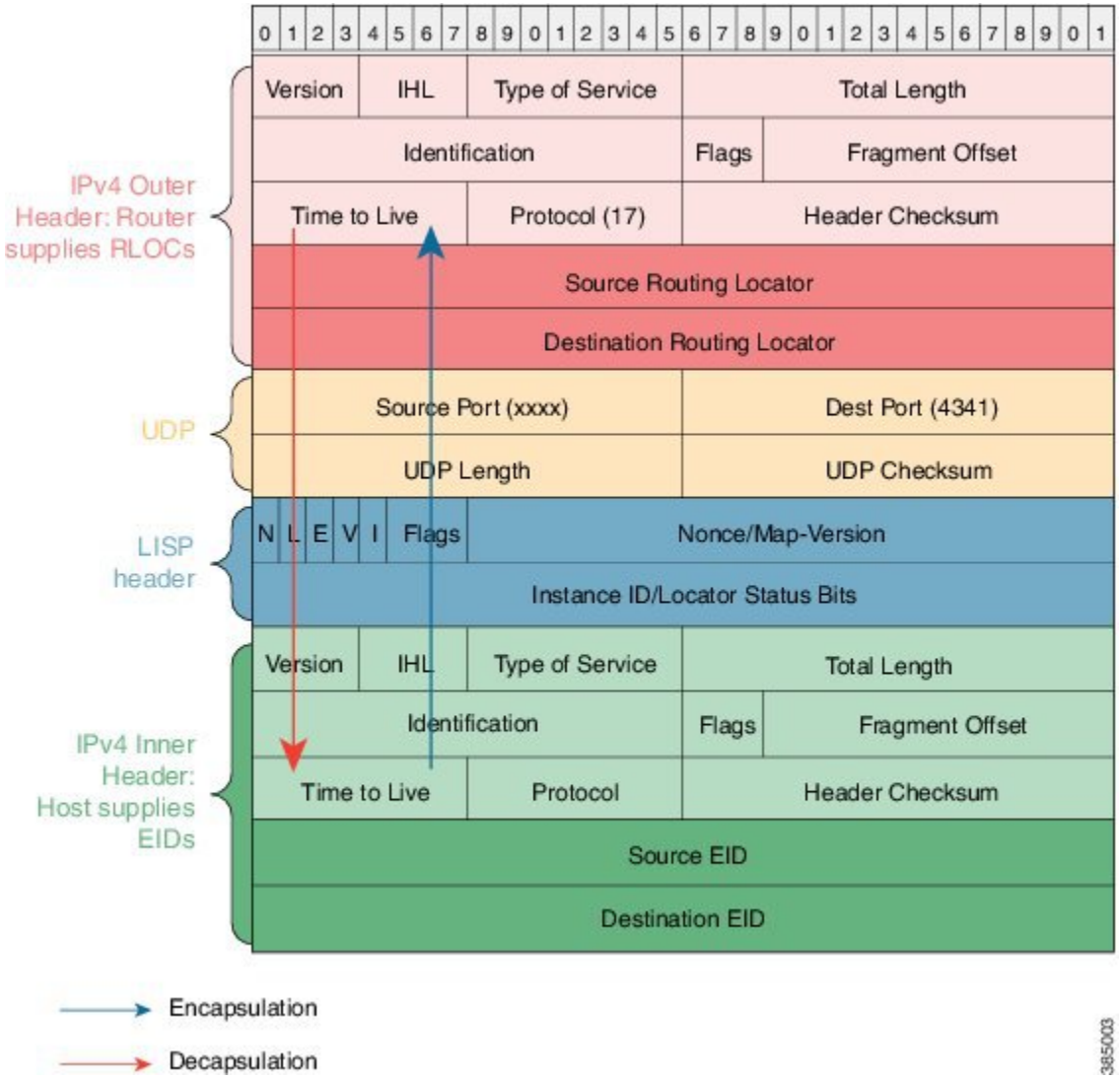
Site ID Qualification

A site is best conceptualized as an authentication domain: A set of ETRs under the same administrative control. The map server authenticates all ETRs in a site using the same shared key. Without the concept of a site, the map server would be required to have prior knowledge of every ETR in the network along with its authentication key. Site managers will not be able to deploy new ETRs without changing the configuration of the map servers. When a site is considered as an authentication domain as opposed to a topological grouping, then it is easy to see that the benefit of site ID qualification resides in the ability of reaching an EID prefix through ETRs under different administrative control.

With Site ID Qualification, the map server can have the same prefix configuration under multiple sites. The name of the feature stems from the requirement that any two sites with at least one prefix in common must be qualified with a unique site IDs.

TTL Propagation

Figure 197: TTL Propagation Mechanism



TTTL Propagation mechanism as shown in the figure is described below:

- A LISP ITR encapsulates a packet and copies TTL value from inner header to outer header.
- A LISP ETR decapsulates a packet and copies TTL value from outer header to inner header if the outer header TTL value is smaller than the inner header TTL.

When TTL propagation is enabled the traceroute tool can display all middle hops between an LISP ITR and ETR. However, when RLOC and EID are of different address-family the traceroute output is undesirable.

When the above cross address-family situation exists, LISP does not propagate TTL between inner and outer IPv4 or IPv6 headers. During encapsulation, ITR uses the maximum permissible TTL in the outer header instead of using the TTL value from the inner header.

It is better to make the LISP tunnel between the ITR and ETR appear as a single hop to the client of traceroute. This is done through the disable-ttl-propagate configuration CLI either for a specific eid-table or the entire router lisp tag.

Note The TTL propagation is turned on automatically.

How to Configure Site ID Qualification

Configuring Site ID Qualification

```
site A
    conf t
router lisp
site A
site-id 1
authentication-key key1
eid-prefix 1.2.0.0/16 accept-more-specifics


site B
   conf t
router lisp
site A
site-id 1
authentication-key key2
eid-prefix 1.2.0.0/16 accept-more-specifics
```

Example: Site ID Qualification

When a site ID registration is received, the map server searches for the longest matching configured prefix. If the resulting prefix is less specific than the registration and does not have "accept-more-specifics" keyword, the registration is rejected; otherwise it is authenticated using the key of the site associated with the prefix. In this example "lazy" map server configuration is used so that an ETR can register any prefix with the map server.

Lazy Map Server Configuration:

Note Setup a new MSMR that has the same lazy configuration for two different sites.

```
enable
conf t
router lisp
 locator-table default
 site A
  site-id 100
  authentication-key key1
  eid-prefix 0.0.0.0/0 accept-more-specifics
  eid-prefix 2000:AAAA:BBBB::/96 accept-more-specifics
  exit
```

```
!
site B
 site-id 200
 authentication-key key2
 eid-prefix 0.0.0.0/0 accept-more-specifics
 eid-prefix 10.0.0.0/8 accept-more-specifics
 eid-prefix 21.0.0.0/8 accept-more-specifics
 eid-prefix 2000:BBBB:AAAA::/96 accept-more-specifics
 exit
ipv4 map-server
ipv4 map-resolver
ipv6 map-server
ipv6 map-resolver
exit
```

How to Disable TTL Propagation

Note The TTL propagation can be disabled for a specific EID-table or an entire router LISP tag.

Disabling TTL Propagation for EID-Table

```
enable
configure terminal
router lisp
eid-table default instance-id 0
disable-ttl-propagate
end
```

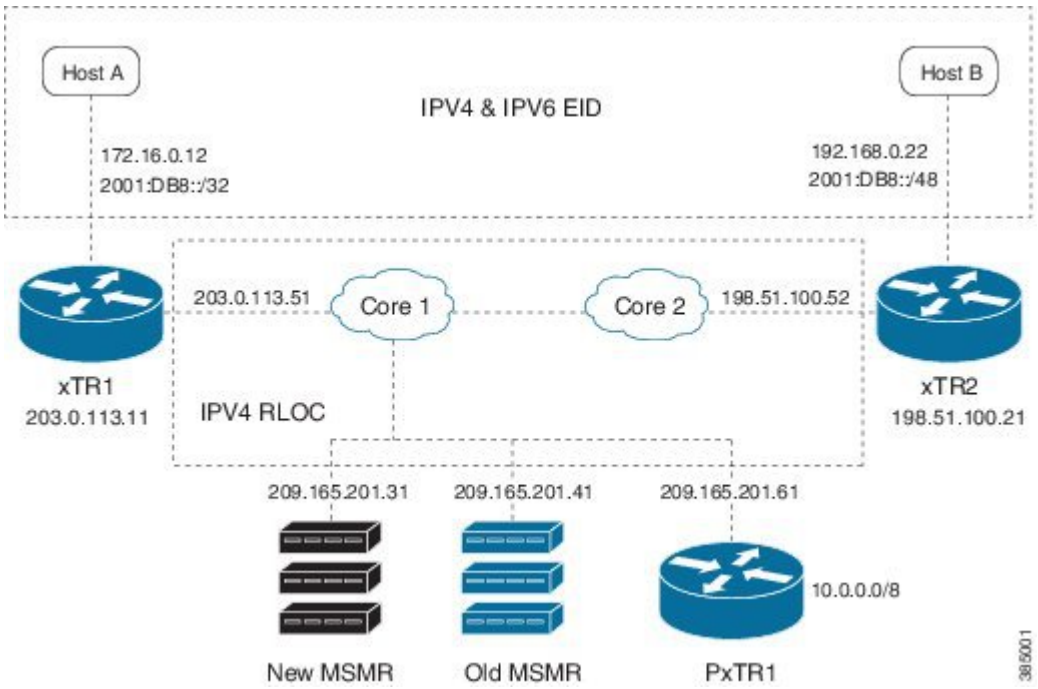
Disabling TTL Propagation for Router LISP Tag

```
enable
configure terminal
router lisp
disable-ttl-propagate
end
```

Verifying TTL Propagate Disable

Perform this task to verify the TTL Propagate Disable feature which is enabled automatically in the LISP network. In this example, a LISP site uses a single edge router that functions as both ITR and ETR (known as an xTR). Routing Locators (RLOCs) are in IPv4. EID prefixes are in both IPv4 and IPv6. The LISP site registers to two map server/map resolver (MSMR) devices in the network core. The topology used in verifying TTL Propagate Disable is as shown in the figure below.

Figure 198: TTL Propagate Disable Topology



The components as shown in the topology are described below:

- xTR1 and xTR2 are xTRs for 2 LISP sites.
- Core1 and Core 2 are routing locators (RLOCs) core routers with no LISP configuration.
- New MSMR is a map-server and map-resolver with reliable map-registration support, whereas Old MSMR does not support reliable map-registration.
- PxTR1 works as a Proxy Ingress Tunnel Router (PITR) and Proxy Egress Tunnel Router (PETR) between the network with 10.0.0.0/8 prefix and the LISP sites.
- Only static routing protocols are used in this setup to reduce control traffic.

Note An IPv6 EID and IPv4 RLOC traceroute output will hide the middle hops between ITR and ETR even when TTL propagation is not disabled.

After disabling TTL propagation, an IPv4 EID over IPv4 RLOC traceroute output appears as below on Host A:

```
Device# traceroute 192.168.0.22

Type escape sequence to abort.
Tracing the route to 192.168.0.22
VRF info: (vrf in name/id, vrf out name/id)
1 203.0.113.11 1 msec 1 msec 0 msec
2 10.40.40.21 1 msec 1 msec 1 msec
3 192.168.0.22 0 msec 2 msec *
```

Additional References for TTI Propagate Disable and Site-ID Qualification

Related Documents

Document Title	Location
Cisco IOS commands	Cisco IOS Master Command List, All Releases
LISP commands	Cisco IOS IP Routing: LISP Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 6830	*The Locator/ID Separation Protocol (LISP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for TTL Propagate Disable and Site-ID Qualification

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 200: Feature Information for TTL Propagate Disable and Site-ID Qualification

Feature Name	Releases	Feature Information
TTL Propagate Disable and Site-ID Qualification		The TTL Propagate Disable feature supports disabling of the TTL (Time-To-Live) propagation for implementing the traceroute tool in a LISP network when RLOC and EID belong to different address-family. The Site ID Qualification feature supports Endpoint Identifier (EID) prefix registration by multiple LISP sites. The following commands were modified: **disable-ttl-propagate**, **eid-prefix**, **eid-table**, **router lisp**, **site-id**, **traceroute**.

CHAPTER 191

DNA SA Border Node Support

Digital Network Architecture (DNA) Security Access (SA) is an Enterprise architecture that brings together multiple building blocks needed for a programmable, secure, and highly automated fabric. Secure Fabric forms the foundation of this architecture and is targeted to address next generation campus trends. From Cisco IOS XE Everest 16.4.1 release, ASR 1000/ISR 4000 platforms can be supported as the border node of DNA SA fabric, handing off the enterprise campus fabric to iWAN, providing IP connectivity across campus and branches. The fabric is separated for campus and branches, and the border node will hand off the LISP/VxLAN-GPO fabric to WAN. In the 16.4.1 release, the handoff is to the DMVPN/MPLS WAN with manual configuration.

Restrictions for DNA SA Border Node Support

- IPv6 RLOC and IPv6 EID is not supported for DNA SA.
- IPv4 SGT can control (enable or disable) IPv4/IPv6 EID SGT. IPv6 SGT is not supported.
- Multicast configuration cannot change encapsulation type.

Information About DNA SA Border Node Support

Enabling VxLAN Encapsulation for LISP Control Plane

To enable VXLAN encapsulation for LISP, use the encapsulation vxlan command in the router lisp configuration mode. This command must be configured on all LISP edge devices in the enterprise fabric deployment: Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Proxy Ingress Tunnel Router (PITR), Proxy Egress Tunnel Router (PETR). Failure to configure this command on any of the LISP edge devices will result in loss of control and data traffic.

Use the **show platform software lisp udp-src-port ipv4** *src_ip dest_ip protocol* command to see the UDP source port according to the data packets. You can also use **ipv6** in the command.

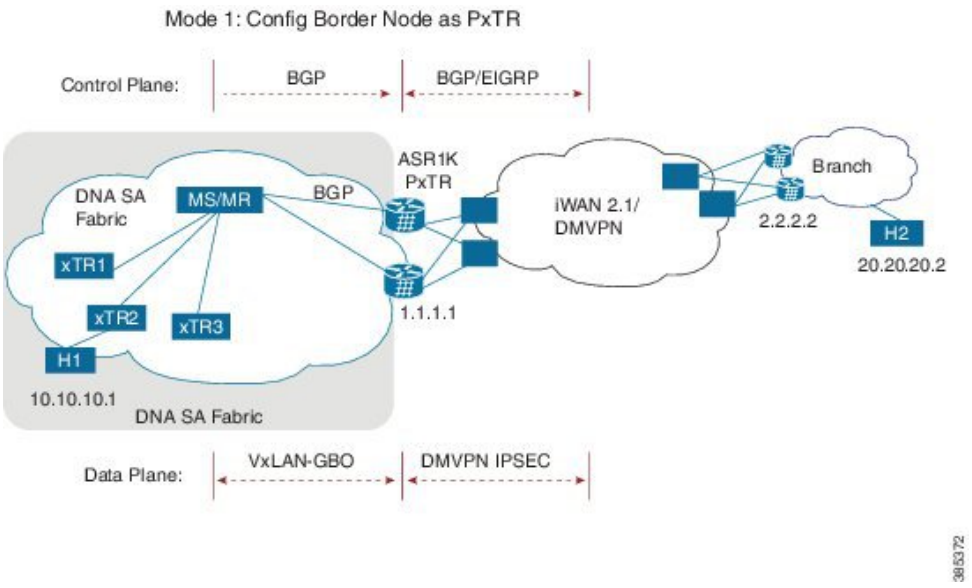
Note VXLAN must not be configuration on the device when VXLAN encapsulation is enabled for LISP. Conversely, VXLAN encapsulation for LISP must not be enabled when configuring other VXLAN protocols.

Two deployment modes are supported, one is to configure border node as PxTR and the other is to configure border node as XTR.

Configuring Border Node as LISP PxTR

Border node can be configured as PxTR for the fabric.

Figure 199: Border Node as LISP PxTR



Control Plane Connectivity

Campus-to-Branches direction:

- xTR will register its direct attached host to MS/MR through LISP map-register.
- There will be per-VRF BGP sessions between MS/MR and PxTR, MS/MR will advertise LISP routes to PxTR
- PxTR will re-originate those routes to WAN through EIGRP or BGP.

Branches-to-Campus direction:

- Branch routes will advertise its routes to border nodes of campus through EIGRP or BGP.
- Border nodes (PxTR) will not advertise routes to LISP MS/MR.
- On XTR, configure “ipv4 use-petr <rloc of PxTR> ”

Packet Flow with Control Plan Interworking

H1 to H2: SIP:10.10.10.1, DIP: 20.20.20.2

- Assuming xTR2 is the default gateway for H1 (it might not be the access switch, but the distribution switch instead). H1 sends the IP packet to xTR2 after it resolves the ARP entry for gateway MAC.

- On xTR2, the IPv4 use-petr 2.2.2.2 is configured.
- On xTR2, a MAP request is initiated to MAP request, to resolve 20.20.20.2
- A negative MAP reply is sent from MS/MR to xTR2.
- xTR2 encapsulation with LISP head and sends to LISP PxTR 1.1.1.1
- Branch router 2.2.2.2 advertises 20.20.20/24 routes to border node 1.1.1.1 using WAN protocol BGP/EIGRP.
- PxTR send the packet to remote branch router 2.2.2.2 through iWAN/DMVPN.

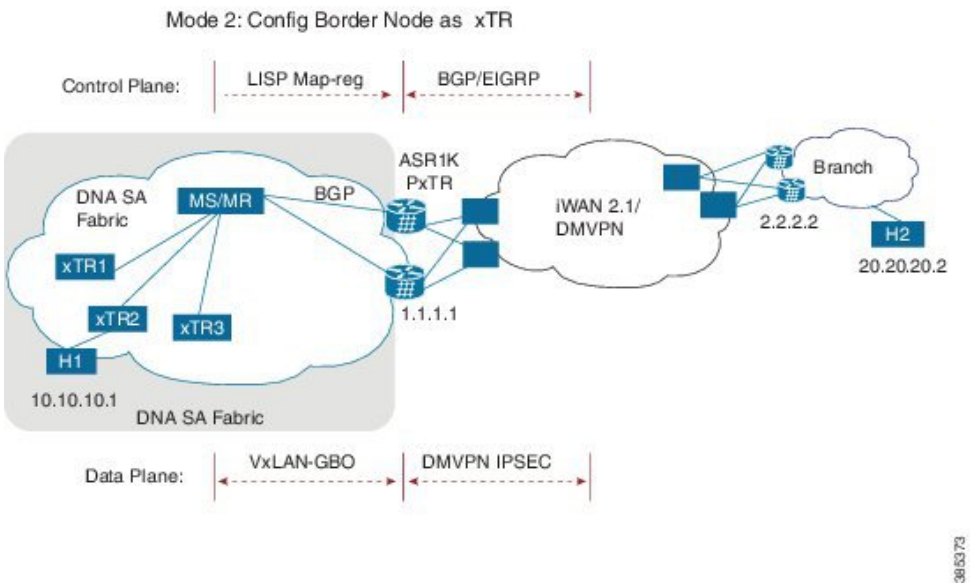
H2 to H1: SIP: 20.20.20.2, DIP: 10.10.10.1

- xTR2 register 10.10.10.1 to MS/MS through LISP MAP-register.
- MS/MR advertise this route to PxTR 1.1.1.1
- PxTR re-originates route to branch route 2.2.2.2
- H2 sends the packets to branch router 2.2.2.2
- Branch router 2.2.2.2 forwards the packets to PxTR 1.1.1.1
- PxTR sends MAP-request to resolve 10.10.10.1, and the MAP-reply is from xTR2.
- PxTR sends LISP packets to xTR2 and then to H1.

Configuring Border Node as LISP xTR

Border node can be configured as xTR for the fabric.

Figure 200: Border Node as LISP xTR



Control Plane Connectivity

Campus-to-Branches direction--For each subnet of fabric, you must manually configure a static route to null0 on ASR1K xTR. Example: ip route vrf vrf1 10.10.10.1 255.255.255.0 Null0 tag 110 ASR1K xTR (1.1.1.1) will advertise this static route to remote branches (2.2.2.2) through BGP or EIGRP.

Branches-to-Campus direction--Remote Branch (2.2.2.2) will advertise routes 20.20.20.2 to ASR1K xTR (1.1.1.1) through BGP or EIGRP. On ASR1K xTR, configure “ipv4 route-import database bgp 100 …” under

LISP EID table to import BGP/EIGRP as LISP EID table. ASR1K xTR 2.2.2.2 will initiate MAP-register to register the EID learnt from BGP.

Packet Flow with Control Plan Interworking

H1 to H2: SIP:10.10.10.1, DIP: 20.20.20.2

- Branch route 2.2.2.2 advertises routes 20.20.20.0/24 to LISP xTR 1.1.1.1 through BGP/EIGRP.
- LISP xTR 1.1.1.1 will import 20.20.20.0/24 into local EID table.
- LISP xTR 1.1.1.1 sends MAP-register to MS/MR to register 20.20.20.0/24 as its local EID
- H1 sends IP packets to xTR2 after it resolves the MAC address of xTR2.
- xTR2 sends map-request to resolve the device for 20.20.20.2 and the RLOC is 1.1.1.1
- xTR2 sends VxLAN encapsulated packets to 1.1.1.1
- RLOC 1.1.1.1 terminates VxLAN and forwards the packets to 2.2.2.2.

H2 to H1: SIP: 20.20.20.2, DIP: 10.10.10.1

- Static route of 10.10.10.1/24 is configured on xTR 1.1.1.1 and it points to null0
- xTR advertises this route to branch 2.2.2.2
- H2 sends packets to branch router 2.2.2.2
- Branch router forwards the packets to LISP xTR 1.1.1.1
- Branch router 2.2.2.2 forwards the packets to PxTR 1.1.1.1
- On LISP xTR 1.1.1.1, 10.10.10.1/24 is pointed to null0, which will trigger LISP routing; it will send MAP-request to resolve the RLOC for 10.10.10.1.
- LISP xTR 1.1.1.1 sends VxLAN encapsulated packets to xTR2.

Security Group Tag (SGT) Propagation

Besides the control plane and data plane connectivity, the SGT tag must be carried over from the campus fabric to WAN and vice-versa, so that SGT tag based policy will be enforced end-to-end across campus and branches. This function has dependence on WAN; if the WAN cannot carry the SGT tag, the tag will be lost.

Configuration Example: Border Node as LISP PxTR

Border node configuration:

```
vrf definition vrf1
 rd 1:1
 !
 address-family ipv4
  route-target export 1:1
  route-target import 1:1
 exit-address-family
 !
vrf definition vrf2
 rd 1:2
!
address-family ipv4
 route-target export 1:2
 route-target import 1:2
exit-address-family
!
interface Loopback1
 vrf forwarding vrf1
```

```
ip address 7.7.7.7 255.255.255.255
!
interface Tunnel100
  description “iwan tunnel for vrf1”
 vrf forwarding vrf1
 ip address 100.0.0.1 255.255.255.0
 tunnel source GigabitEthernet2
 tunnel destination 16.0.0.2
 tunnel key 100
 !
 interface Tunnel101
 description “iwan tunnel for vrf2”
 vrf forwarding vrf2
 ip address 101.0.0.1 255.255.255.0
 tunnel source GigabitEthernet2
 tunnel destination 16.0.0.2
 tunnel key 101
!
interface Tunnel1000
 description “pxtr and msmr tunnel vrf1”
vrf forwarding vrf1
ip address 200.0.0.2 255.255.255.0
tunnel source GigabitEthernet1
tunnel destination 13.0.0.1
tunnel key 1000
!
interface Tunnel1001
 description “pxtr and msmr tunnel vrf2”
vrf forwarding vrf2
ip address 201.0.0.2 255.255.255.0
tunnel source GigabitEthernet1
tunnel destination 13.0.0.1
tunnel key 1001
!
interface GigabitEthernet1
 ip address 15.0.0.2 255.255.255.0
 ip ospf 1 area 0
!
interface GigabitEthernet2
 ip address 16.0.0.1 255.255.255.0
!
router lisp
 encapsulation vxlan    //Enable VXLAN GPO encapsulation for the LISP data plane//
 eid-table default instance-id 0
 map-cache 0.0.0.0/0 map-request
 exit
!
eid-table vrf vrf1 instance-id 1
 ipv4 route-import map-cache bgp 100 route-map set_lisp_vrf1
 exit
!
eid-table vrf vrf2 instance-id 2
 ipv4 route-import map-cache bgp 100 route-map set_lisp_vrf2
 exit
 !
ipv4 sgt     //enable SGT function for SGT tag propagation//
 exit
 !
 ipv4 map-request-source 14.0.0.2
 ipv4 proxy-etr
 ipv4 proxy-itr 15.0.0.2
 ipv4 itr map-resolver 14.0.0.1
 exit
 !
```

```
 router ospf 1
 !
 router bgp 100
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf vrf1
  neighbor 100.0.0.2 remote-as 100
  neighbor 100.0.0.2 activate
  neighbor 200.0.0.1 remote-as 200
  neighbor 200.0.0.1 ebgp-multihop 255
  neighbor 200.0.0.1 update-source Tunnel1000
  neighbor 200.0.0.1 activate
  neighbor 200.0.0.1 send-community both
 exit-address-family
 !
address-family ipv4 vrf vrf2
 neighbor 101.0.0.2 remote-as 100
 neighbor 101.0.0.2 activate
 neighbor 201.0.0.1 remote-as 200
 neighbor 201.0.0.1 ebgp-multihop 255
 neighbor 201.0.0.1 update-source Tunnel1001
 neighbor 201.0.0.1 activate
 neighbor 201.0.0.1 send-community both
exit-address-family
!
ip bgp-community new-format
ip community-list 10 permit 1000:1
ip community-list 11 permit 1000:2
!
route-map set_lisp_vrf1 permit 10
match community 10
!
route-map set_lisp_vrf2 permit 10
match community 11
!
!
MSMR configuration:
vrf definition vrf1
 rd 1:1
 !
 address-family ipv4
 exit-address-family
!
vrf definition vrf1000
 rd 1000:1
 !
address-family ipv4
exit-address-family
!
vrf definition vrf2
 rd 1:2
 !
address-family ipv4
exit-address-family
!
interface Loopback0
 ip address 14.0.0.1 255.255.255.255
 ip ospf 1 area 0
 !
interface Tunnel1000
 description "pxtr and msmr tunnel vrf1"
 vrf forwarding vrf1
 ip address 200.0.0.1 255.255.255.0
 tunnel source GigabitEthernet3.6
```

```
 tunnel destination 15.0.0.2
 tunnel key 1000
!
interface Tunnel1001
 description “pxtr and msmr tunnel vrf2”
 vrf forwarding vrf2
 ip address 201.0.0.1 255.255.255.0
 tunnel source GigabitEthernet3.6
 tunnel destination 15.0.0.2
 tunnel key 1001
!
interface GigabitEthernet2
 no ip address
!
interface GigabitEthernet2.4
 encapsulation dot1Q 4
 ip address 12.0.0.2 255.255.255.0
 ip ospf 1 area 0
 !
interface GigabitEthernet2.5
 encapsulation dot1Q 5
 ip address 12.0.1.2 255.255.255.0
 ip ospf 1 area 0
 !
interface GigabitEthernet3
 no ip address
 negotiation auto
 cdp enable
!
interface GigabitEthernet3.6
 encapsulation dot1Q 6
 ip address 13.0.0.1 255.255.255.0
 ip ospf 1 area 0
!
interface GigabitEthernet3.7
 encapsulation dot1Q 7
 ip address 13.0.1.1 255.255.255.0
 ip ospf 1 area 0
!
router lisp
 eid-table default instance-id 0
  exit
 !
eid-table vrf vrf1 instance-id 1
 ipv4 route-export site-registrations
 exit
!
eid-table vrf vrf2 instance-id 2
 ipv4 route-export site-registrations
 exit
!
rtr-set rtr
 12.0.0.1 authentication-key cisco
 12.0.1.1 authentication-key cisco
 exit
!
map-server advertise-rtr-set rtr
site xtr1
 authentication-key cisco
 advertise-rtr-set rtr
 eid-prefix 1.1.1.1/32 route-tag 110
 eid-prefix instance-id 1 5.5.5.5/32 route-tag 100
 exit
!
```

```
site xtr2
 authentication-key cisco
 eid-prefix 2.2.2.2/32 route-tag 110
 eid-prefix instance-id 1 6.6.6.6/32 route-tag 100
 eid-prefix instance-id 1 11.11.11.11/32 route-tag 120
 eid-prefix instance-id 2 6.6.6.6/32 route-tag 110
 exit
!
ipv4 map-server
ipv4 map-resolver
exit
!
router ospf 1
!
router bgp 200
 bgp log-neighbor-changes
!
address-family ipv4 vrf vrf1
 redistribute lisp metric 11 route-map set_lisp_vrf1
 neighbor 200.0.0.2 remote-as 100
 neighbor 200.0.0.2 ebgp-multihop 255
 neighbor 200.0.0.2 update-source Tunnel1000
 neighbor 200.0.0.2 activate
 neighbor 200.0.0.2 send-community both
exit-address-family
!
address-family ipv4 vrf vrf2
 redistribute lisp metric 11 route-map set_lisp_vrf2
 neighbor 201.0.0.2 remote-as 100
 neighbor 201.0.0.2 ebgp-multihop 255
 neighbor 201.0.0.2 update-source Tunnel1001
 neighbor 201.0.0.2 activate
 neighbor 201.0.0.2 send-community both
exit-address-family
!
!
ip bgp-community new-format
!
route-map set_lisp_vrf1 permit 10
 match tag 100
 set community 1000:1
!
route-map set_lisp_vrf2 permit 10
 match tag 110
 set community 1000:2
!
```

Configuration Example: Border Node as LISP xTR

Border node configuration:

```
vrf definition vrf1
 rd 1:1
 !
 address-family ipv4
  route-target export 1:1
  route-target import 1:1
 exit-address-family
!
vrf definition vrf2
 rd 1:2
```

```
 !
 address-family ipv4
 exit-address-family
 !
 interface Loopback0
 ip address 2.2.2.2 255.255.255.255
 !
 interface Loopback1
  vrf forwarding vrf1
  ip address 6.6.6.6 255.255.255.255
  !
  interface Tunnel200
  description "iWAN tunnel to remote branch"
   vrf forwarding vrf1
   ip address 150.0.0.2 255.255.255.0
   tunnel source GigabitEthernet2
   tunnel destination 17.0.0.1
   tunnel key 200
 !
 interface GigabitEthernet2
  ip address 17.0.0.2 255.255.255.0
  !
 interface GigabitEthernet3
  no ip address
  !
 interface GigabitEthernet3.6
  encapsulation dot1Q 6
  ip address 13.0.0.2 255.255.255.0
  ip ospf 1 area 0
  !
 interface GigabitEthernet3.7
  encapsulation dot1Q 7
  ip address 13.0.1.2 255.255.255.0
  ip ospf 1 area 0
  !
 interface GigabitEthernet4
  ip address 15.0.0.1 255.255.255.0
  ip ospf 1 area 0
  !
 router lisp
  encapsulation vxlan
  locator-set set1
  13.0.0.2 priority 1 weight 1
  13.0.1.2 priority 1 weight 1
 exit
 !
 eid-table default instance-id 0
  database-mapping 2.2.2.2/32 locator-set set1
  exit
  !
 eid-table vrf vrf1 instance-id 1
  database-mapping 6.6.6.6/32 locator-set set1
  ipv4 route-import database bgp 100 route-map match_com locator-set set1
  exit
  !
 eid-table vrf vrf2 instance-id 2
  database-mapping 6.6.6.6/32 locator-set set1
  exit
  !
 ipv4 sgt    //enable SGT function for SGT tag propagation//
  exit
 !
ipv4 use-petr 15.0.0.2
ipv4 itr map-resolver 14.0.0.1
```

```
ipv4 itr
ipv4 etr map-server 14.0.0.1 key cisco
ipv4 etr
exit
!
router ospf 1
!
router bgp 100
 bgp log-neighbor-changes
!
address-family ipv4 vrf vrf1
 redistribute static route-map tag_110
 neighbor 150.0.0.1 remote-as 100
 neighbor 150.0.0.1 activate
 neighbor 150.0.0.1 send-community both
 exit-address-family

ip bgp-community new-format
ip community-list 10 permit 200:1
ip route vrf vrf1 5.5.5.5 255.255.255.255 Null0 tag 110
!
route-map tag_110 permit 10
 match tag 110
!
route-map match_com permit 10
 match community 10
!
```

Feature Information for DNA SA Border Node Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 201: Feature Information for DNA SA Border Node Support

Feature Name	Releases	Feature Information
DNA SA Border Node Support	Cisco IOS XE Everest 16.4.1 Release	From Cisco IOS XE Everest 16.4.1 release, ASR 1000/ISR 4000 platforms can be supported as the border node of DNA SA fabric, handing off the enterprise campus fabric to iWAN, providing IP connectivity across campus and branches.

CHAPTER 192

LISP Support for TCP Authentication Option

In a LISP deployment, an Egress Tunnel Router (ETR) and a Map Server (MS) exchange LISP control messages through a TCP connection. From Cisco IOS XE Amsterdam 17.2.1, you can use TCP Authentication Option (TCP-AO) to guard against spoofed TCP segments in the sessions between an ETR and an MS.

If you do not configure TCP-AO-based authentication, the TCP segments exchanged between an ETR and an MS are authenticated using a shared key. Cisco IOS XE Amsterdam 17.1.x and earlier releases support only shared-key authentication of TCP messages.

LISP Support for TCP Authentication Option

Overview of LISP Support for TCP Authentication Option

When an ETR detects the first local EID entry, the ETR sends a UDP map-registration message to the MS. MS authenticates the UDP message using a shared authentication key configured on both the ETR and the MS. On successful authentication, the MS creates a TCP listening socket to the ETR for future message exchange.

From Cisco IOS XE Amsterdam 17.2.1, the initial UDP message sent by an ETR also indicates the AO capability.

- If you configure TCP-AO authentication on an ETR, the ETR sets a TCP-AO flag in the UDP message. When the MS detects the TCP-AO flag set in the UDP message, the MS creates the TCP-AO-enabled connection to the ETR. TCP segments that are subsequently exchanged are authenticated using TCP-AO.

  If you do not configure TCP-AO authentication on an ETR, the ETR does not set the TCP-AO flag in the UDP message. When the MS finds that the TCP-AO flag is not set in the UDP message, the MS creates a non-TCP-AO connection to the ETR. TCP messages that are subsequently exchanged are authenticated using the shared key.

- If an ETR is not upgraded to Cisco IOS XE Amsterdam 17.2.1, the ETR sends a UDP message without the TCP-AO flag. When the MS finds no TCP-AO flag in the UDP message, the MS creates a non-TCP-AO connection to the ETR. TCP messages that are subsequently exchanged are authenticated using the shared key.

If you configure peer address locator on a TCP-AO-enabled MS using **map-server session passive-open** [*RLOC*], the MS creates TCP-AO enabled listening sockets. The `accept-ao-mismatch` tcb flag is set to TRUE to maintain backward compatibility with a non-upgraded BR that is not upgraded to Cisco IOS XE Amsterdam 17.2.1 or a later release, or does not have TCP-AO enabled.

Restrictions for LISP Support for TCP Authentication Option

- If you configure TCP-AO authentication on an ETR, but do not configure TCP-AO authentication on the peer MS, a TCP connection is not established between the ETR and the MS. For TCP-AO authentication, you must configure the feature on both the ETR and the MS.
- If you configure or remove TCP-AO authentication for an ETR-MS peer session, the corresponding LISP session flaps as the configuration takes effect.
- You can configure an ETR to use different TCP configurations when communicating with different MSs. When multiple ETRs connect to the same MS, you can configure the ETRs to use different TCP configurations.

  However, on all multi-homing ETRs that are connected to an MS, configure identical TCP-AO behavior. A configuration in which TCP-AO is enabled on some of the multi-homing ETRs connected to the MS but disabled on the others is not supported.

  In a deployment with multiple MSs, a multi-homing ETR can simultaneously have TCP-AO sessions with some of MSs and non-TCP-AO sessions with other MSs. However, all the multi-homing ETRs connected to an MS must have either TCP-AO sessions or non-TCP-AO sessions with the MS.

How to Configure LISP Support for TCP Authentication Option

To establish TCP-AO-based connections between an ETR and an MS, you must configure the following:

1. TCP key-chain and keys on both the ETR and the MS
2. TCP-AO on the MS
3. TCP-AO on the ETR

Configure TCP Key Chain and Keys

Configure TCP-AO key chain and keys on both the peers communicating through a TCP connection.

Note

- Ensure that the key-string, send-lifetimes, cryptographic-algorithm, and ids of keys match on both peers.
- Ensure that the send-id on a router matches the recv-id on the peer router. We recommend using the same id for both the parameters unless there is a need to use separate key spaces.
- The send-id and recv-id of a key cannot be reused for another key in the same key chain.
- Do not modify properties of a key in use, except when you need to modify the send-lifetime of the key to trigger rollover. Before modifying properties other than send-lifetime, disassociate the key from the TCP connection.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **key chain** *key-chain-name* **tcp**

Example:

```
Device(config)# key chain kc1 tcp
```

Creates a TCP-AO key chain of with a specified name and enters the TCP-AO key chain configuration mode.

The key chain name can have a maximum of 256 characters.

Step 4 **key** *key-id*

Example:

```
Device(config-keychain-tcp)# key 10
```

Creates a key with the specified key-id and enters the TCP-AO key chain key configuration mode.

The key-id must be in the range from 0 to 2147483647.

Note The key-id has only local significance. It is not part of the TCP Authentication Option.

Step 5 **send-id** *send-identifier*

Example:

```
Device(config-keychain-tcp-key)# send-id 218
```

Specifies the send identifier for the key.

The send-identifier must be in the range from 0 to 255.

Step 6 **recv-id** *receiver-identifier*

Example:

```
Device(config-keychain-tcp-key)# recv-id 218
```

Specifies the receive identifier for the key.

The receive-identifier must be in the range from 0 to 255.

Step 7 **cryptographic-algorithm** {**aes-128-cmac** | **hmac-sha-1** | **hmac-sha-256**}

Example:

```
Device(config-keychain-tcp-key)# cryptographic-algorithm hmac-sha-1
```

Specifies the algorithm to be used to compute MACs for TCP segments.

aes-128-cmac	AES-128-CMAC-96: Configures AES-128-CMAC as a cryptographic algorithm with a digest size of 12 bytes.
hmac-sha-1	HMAC-SHA1-96: Configures HMAC-SHA1-96 as a cryptographic algorithm with a digest size of 12 bytes.
hmac-sha-256	HMAC-SHA-256: Configures HMAC-SHA-256 as a cryptographic algorithm with a digest size of 32 bytes.

Step 8 (Optional) **include-tcp-options**

Example:

```
Device(config-keychain-tcp-key)# include-tcp-options
```

This flag indicates whether TCP options other than TCP-AO must be used to calculate MACs.

With the flag enabled, the content of all options, in the order present, is included in the MAC and TCP-AO's MAC field is zero-filled.

When the flag is disabled, all options other than TCP-AO are excluded from MAC calculations.

By default, this flag is disabled.

Step 9 **send-lifetime** [**local**] *start-time* {**infinite** | *end-time* | **duration** *seconds*}

Example:

```
Device(config-keychain-tcp-key)# send-lifetime local 12:00:00 28 Feb 2018 duration 20
```

Specifies the time for which the key is valid to be used for TCP-AO authentication in the send direction.

Use the **local** keyword to specify the start-time in the local time zone. By default, the start-time corresponds to UTC time.

Step 10 **key-string** *master-key*

Example:

```
Device(config-keychain-tcp-key)# key-string abcde
```

Specifies the primary-key for deriving traffic keys.

The primary-keys must be identical on both the peers. If the primary-keys do not match, authentication fails and segments may be rejected by the receiver.

Step 11 (Optional) **accept-ao-mismatch**

Example:

```
Device(config-keychain-tcp-key)# accept-ao-mismatch
```

This flag indicates whether the receiver should accept segments for which the MAC in the incoming TCP AO does not match the MAC generated on the receiver.

Note Use this configuration with caution. This configuration disables TCP-AO functionality and key rollover on associated connections.

Step 12 **end**

Example:

```
Device(config-keychain-tcp-key)# end
```

Exits TCP-AO key chain key configuration mode and returns to privileged EXEC mode.

Configure TCP Authentication Option on MS

To configure TCP-AO on an MS,

- configure the key chain that the MS uses for TCP-AO authentication of segments received from peer ETRs. Use the command **tcp auth-option** *keychain-name* in the LISP configuration (`config-router-lisp`) mode.

  The MS uses a common key chain for all peer ETRs.

- configure the MS to accept TCP-AO-based connection requests from peer ETRs. Use the **peer accept** command in the TCP-AO(`tcp auth-option`) configuration mode.

- configure the shared authentication key that the MS uses to authenticate the initial UDP message from a peer ETR. The MS authenticates the TCP segments from the ETR using the same key if TCP-AO is not configured on the ETR.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router lisp**

Example:

```
Device(config)# router lisp
```

Enters LISP configuration mode.

Step 4 **tcp auth-option** *key-chain*

Example:

```
Device(config-router-lisp)# tcp auth-option kc1
```

Configures the MS to use the specified key chain for TCP-AO and enters the TCP-AO configuration mode.

Step 5 **peer accept**

Example:

```
Device(config-router-lisp-tcp-ao)# peer accept
```

Configures the MS to accept TCP-AO-based connection requests from peer ETRs.

Step 6 **exit**

Example:

```
Device(config-router-lisp-tcp-ao)# exit
```

Exits the TCP-AO configuration mode and returns to LISP configuration mode.

Step 7 **service ipv4**

Example:

```
Device(config-router-lisp)# service ipv4
```

Configures IPv4 as a service type and enters LISP service IPv4 configuration mode.

Step 8 **map-server**

Example:

```
Device(config-lisp-srv-ipv4)# map-server
```

Enables LISP map server functionality for EIDs in the IPv4 address family.

Step 9 **map-resolver**

Example:

```
Device(config-lisp-srv-ipv4)# map-resolver
```

Enables LISP map resolver functionality for EIDs in the IPv4 address family.

Step 10 **exit-service-ipv4**

Example:

```
Device(config-lisp-srv-ipv4)# exit-service-ipv4
```

Exits LISP service IPv4 configuration mode and returns to LISP configuration mode.

Step 11 **site** *site-name*

Example:

```
Device(config-router-lisp)# site site-a
```

Specifies a LISP site named site-a and enters LISP site configuration mode.

Step 12 **eid-record instance-id** *instanace-id EID-prefix*

Example:

```
Device(config-router-lisp-site)# eid-record instance-id 1 172.16.1.0/24
```

Configures an IPv4 prefix associated with this LISP site.

Repeat this step as necessary to configure additional EID prefixes under this LISP sites.

Note The LISP ETR must be configured with matching EID prefixes.

Step 13 **authentication-key** [*key-type*] *authentication-key*

Example:

```
Device(config-router-lisp-site)# authentication-key some-key
```

Configures the authentication key associated with this site.

Note Configure an identical authentication key on the LISP ETR.

Step 14 **exit-site**

Example:

```
Device(config-router-lisp-site)# exit-site
```

Exits LISP site configuration mode and returns to LISP configuration mode.

Step 15 Repeat Steps 11 through 14 to configure additional LISP sites.

Configure TCP Authentication Option on ETR

To configure TCP-AO on an ETR,

- configure the key chain that the ETR uses for TCP-AO authentication of segments received from an MS. Use the command **tcp auth-option** *keychain-name* in the LISP configuration (`config-router-lisp`) mode.

  An ETR may use different key chains for different MS peers.

- configure the ETR to establish TCP-AO connection with an MS peer. Use the **peer** *map-server-address* command in the TCP-AO(`tcp auth-option`) configuration mode.

- configure the shared authentication key that the ETR uses to register with the MS.

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **router lisp**

Example:

```
Device(config)# Device lisp
```

Enters LISP configuration mode.

Step 4 **tcp auth-option** *key-chain*

Example:

```
Device(config-router-lisp)# tcp auth-option kc1
```

Configures the ETR to use the specified key chain for TCP-AO and enters the TCP-AO configuration mode.

Step 5 **peer** *map-server-address*

Example:

```
Device(config-router-lisp-tcp-auth-option)# peer 10.10.10.10
```

Configures the ETR to establish a TCP-AO-based connection with the specified MS.

Step 6 **exit**

Example:

```
Device(config-router-lisp-tcp-auth-option)# exit
```

Exits the TCP-AO configuration mode and returns to LISP configuration mode.

Step 7 **service ipv4**

Example:

```
Device(config-router-lisp)# service ipv4
```

Configures IPv4 as a service type and enters LISP service IPv4 configuration mode.

Step 8 **etr**

Example:

```
Device(config-lisp-srv-ipv4)# etr
```

Enables LISP ETR functionality for the IPv4 address family.

Step 9 **map-server** *map-server-address* **key** [*key-type*] *authentication-key*

Example:

```
Device(config-lisp-srv-ipv4)# map-server 10.10.10.10 key some-key
```

Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system.

Step 10 **exit-service-ipv4**

Example:

```
Device(config-lisp-srv-ipv4)# exit-service-ipv4
```

Exits LISP service IPv4 configuration mode and returns to LISP configuration mode.

Verifying LISP Support for TCP Authentication Option

Use the command **show lisp vrf default session** *peer-address* to verify that TCP-AO is enabled and the correct TCP key chain is in use.

The following example shows the output of the **show lisp vrf default session** *peer-address* command. The highlighted line is included in the output only if TCP-AO is enabled.

```
Router#show lisp vrf default session 4.4.4.4

Peer address:     4.4.4.4:4342
Local address:    2.2.2.2:34316
Session Type:     Active
Session State:    Up (00:01:12)
Messages in/out:  6/5
Bytes in/out:     245/292
Fatal errors:     0
```

```
Rcvd unsupported: 0
Rcvd invalid VRF: 0
Rcvd override:    0
Rcvd malformed:   0
Sent deferred:    0
SSO redundancy:   unsynchronized
Auth type:        TCP-Auth-Option, keychain: kc1

Accepting Users:  0
Users:            6
  Type                      ID                                      In/Out    State
  ETR Reliable Registration lisp 0 IID 102 AFI IPv4                  2/2      TCP
  ETR Reliable Registration lisp 0 IID 108 AFI IPv4                  2/2      TCP
  Capability Exchange       N/A
```

Debugging LISP Support for TCP Authentication Option

You can use the following commands to debug TCP-AO operation with LISP:

- **debug lisp control-plane session**
- **debug ip tcp mkt**
- **debug ip tcp transactions**

Additional References

Related Documents

Related Topic	Document Title
TCP Authentication Option	IP Routing: Protocol-Independent Configuration Guide

Standards and RFCs

Standard/RFC	Title
RFC 5925	The TCP Authentication Option

PART VII

OSPF

CHAPTER 193

Configuring OSPF

This module describes how to configure Open Shortest Path First (OSPF). OSPF is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). OSPF was designed expressly for IP networks and it supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

Cisco supports RFC 1253, *OSPF Version 2 Management Information Base*, August 1991. The OSPF MIB defines an IP routing protocol that provides management information related to OSPF and is supported by Cisco routers.

For protocol-independent features that work with OSPF, see the "Configuring IP Routing Protocol-Independent Features" module.

Information About OSPF

Cisco OSPF Implementation

The Cisco implementation conforms to the OSPF Version 2 specifications detailed in the Internet RFC 2328. The following list outlines key features supported in the Cisco OSPF implementation:

- Stub areas—The definition of stub areas is supported.
- Route redistribution—Routes learned via any IP routing protocol can be redistributed into any other IP routing protocol. At the intradomain level, OSPF can import routes learned via Interior Gateway Routing Protocol (IGRP), Routing Information Protocol (RIP), and Intermediate System-to-Intermediate System (IS-IS). OSPF routes can also be exported into IGRP, RIP, and IS-IS. At the interdomain level, OSPF can import routes learned via Exterior Gateway Protocol (EGP) and Border Gateway Protocol (BGP). OSPF routes can be exported into EGP and BGP.
- Authentication—Plain text and message-digest algorithm 5 (MD5) authentication among neighboring routers within an area is supported.

- Routing interface parameters—Configurable parameters supported include interface output cost, retransmission interval, interface transmit delay, router priority, router "dead" and hello intervals, and authentication key.
- Virtual links—Virtual links are supported.
- Not-so-stubby area (NSSA)—RFC 3101, which replaces and is backward compatible with RFC 1587.
- OSPF over demand circuit—RFC 1793.

Note From Cisco IOS XE 17.13.1a, if you change the router ID (RID) for an OSPF instance even with an active adjacency, the new RID changes take effect immediately. There is no need to issue a **reload** command or **clear ip ospf** process command . If OSPF adjacencies are up, they will also be reset with the new RID.

Router Coordination for OSPF

OSPF typically requires coordination among many internal routers: Area Border Routers (ABRs), which are routers connected to multiple areas, and Autonomous System Boundary Routers (ASBRs). At a minimum, OSPF-based routers or access servers can be configured with all default parameter values, no authentication, and interfaces assigned to areas. If you intend to customize your environment, you must ensure coordinated configurations of all routers.

Route Distribution for OSPF

You can specify route redistribution; see the task "Redistribute Routing Information" in the *Network Protocols Configuration Guide, Part 1*, for information on how to configure route redistribution.

The Cisco OSPF implementation allows you to alter certain interface-specific OSPF parameters, as needed. You are not required to alter any of these parameters, but some interface parameters must be consistent across all routers in an attached network. Those parameters are controlled by the **ip ospf hello-interval**, **ip ospf dead-interval**, and **ip ospf authentication-key** interface configuration commands. Therefore, if you do configure any of these parameters, ensure that the configurations for all routers on your network have compatible values.

By default, OSPF classifies different media into the following three types of networks:

- Broadcast networks (Ethernet, Token Ring, and FDDI)
- Nonbroadcast multiaccess (NBMA) networks (Switched Multimegabit Data Service [SMDS], Frame Relay, and X.25)
- Point-to-point networks (High-Level Data Link Control [HDLC] and PPP)

You can configure your network as either a broadcast or an NBMA network.

X.25 and Frame Relay provide an optional broadcast capability that can be configured in the map to allow OSPF to run as a broadcast network. See the **x25 map** and **frame-relay map** command pages in the *Cisco IOS Wide-Area Networking Command Reference* publication for more detail.

OSPF Network Type

You have the choice of configuring your OSPF network type as either broadcast or NBMA, regardless of the default media type. Using this feature, you can configure broadcast networks as NBMA networks when, for example, you have routers in your network that do not support multicast addressing. You also can configure NBMA networks (such as X.25, Frame Relay, and SMDS) as broadcast networks. This feature saves you from needing to configure neighbors, as described in the "Configuring OSPF for Nonbroadcast Networks"section later in this module.

Configuring NBMA networks as either broadcast or nonbroadcast assumes that there are virtual circuits (VCs) from every router to every router, that is, a fully meshed network. This is not true in some cases, for example, because of cost constraints or when you have only a partially meshed network. In these cases, you can configure the OSPF network type as a point-to-multipoint network. Routing between two routers that are not directly connected will go through the router that has VCs to both routers. Note that you need not configure neighbors when using this feature.

An OSPF point-to-multipoint interface is defined as a numbered point-to-point interface having one or more neighbors. It creates multiple host routes. An OSPF point-to-multipoint network has the following benefits compared to NBMA and point-to-point networks:

- Point-to-multipoint is easier to configure because it requires no configuration of neighbor commands, it consumes only one IP subnet, and it requires no designated router election.
- It costs less because it does not require a fully meshed topology.
- It is more reliable because it maintains connectivity in the event of VC failure.

On point-to-multipoint broadcast networks, there is no need to specify neighbors. However, you can specify neighbors with the **neighbor** router configuration command, in which case you should specify a cost to that neighbor.

Before the **point-to-multipoint** keyword was added to the **ip ospf network** interface configuration command, some OSPF point-to-multipoint protocol traffic was treated as multicast traffic. Therefore, the **neighbor** router configuration command was not needed for point-to-multipoint interfaces because multicast took care of the traffic. Hello, update, and acknowledgment messages were sent using multicast. In particular, multicast hello messages discovered all neighbors dynamically.

On any point-to-multipoint interface (broadcast or not), the Cisco IOS software assumed that the cost to each neighbor was equal. The cost was configured with the **ip ospf cost** interface configuration command. In reality, the bandwidth to each neighbor is different, so the cost should differ. With this feature, you can configure a separate cost to each neighbor. This feature applies to point-to-multipoint interfaces only.

Because many routers might be attached to an OSPF network, a *designated router* is selected for the network. Special configuration parameters are needed in the designated router selection if broadcast capability is not configured.

These parameters need only be configured in those devices that are themselves eligible to become the designated router or backup designated router (in other words, routers with a nonzero router priority value).

You can specify the following neighbor parameters, as required:

- Priority for a neighboring router
- Nonbroadcast poll interval

On point-to-multipoint, nonbroadcast networks, use the **neighbor** router configuration command to identify neighbors. Assigning a cost to a neighbor is optional.

Prior to Cisco IOS Release 12.0, some customers were using point-to-multipoint on nonbroadcast media (such as classic IP over ATM), so their routers could not dynamically discover their neighbors. This feature allows the **neighbor** router configuration command to be used on point-to-multipoint interfaces.

Area Parameters

Use OSPF Not-So-Stubby Areas (NSSA) feature to simplify administration if you are an Internet service provider (ISP) or a network administrator that must connect a central site that is using OSPF to a remote site that is using a different routing protocol.

Prior to NSSA, the connection between the corporate site border router and the remote router could not be run as an OSPF stub area because routes for the remote site could not be redistributed into the stub area, and two routing protocols needed to be maintained. A simple protocol such as RIP was usually run and handled the redistribution. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

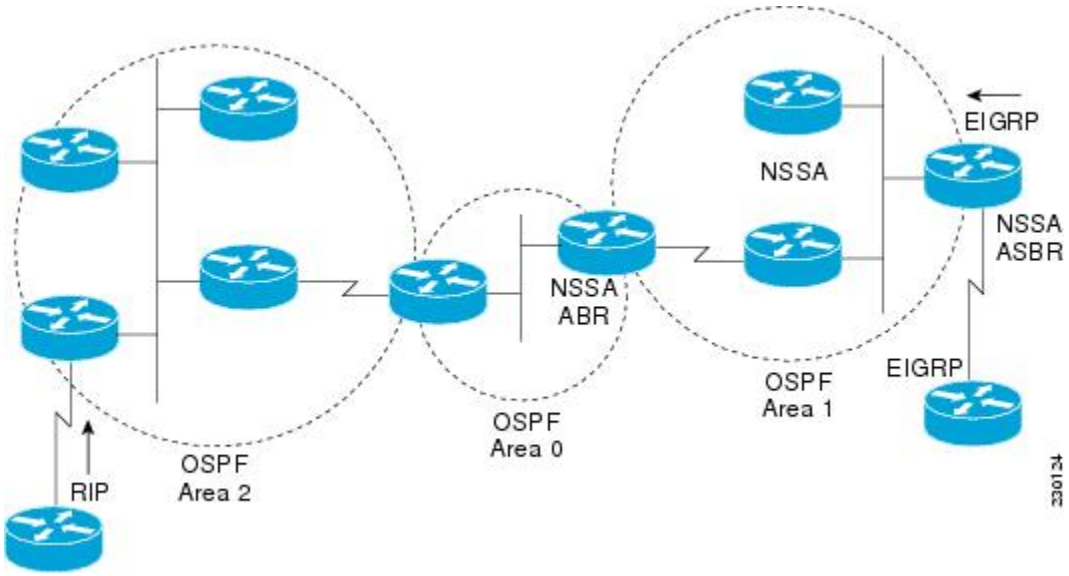
As with OSPF stub areas, NSSA areas cannot be injected with distributed routes via Type 5 LSAs. Route redistribution into an NSSA area is possible only with a special type of LSA that is known as Type 7 that can exist only in an NSSA area. An NSSA ASBR generates the Type 7 LSA so that the routes can be redistributed, and an NSSA ABR translates the Type 7 LSA into a Type 5 LSA, which can be flooded throughout the whole OSPF routing domain. Summarization and filtering are supported during the translation.

RFC 3101 allows you to configure an NSSA ABR router as a forced NSSA LSA translator. This means that the NSSA ABR router will unconditionally assume the role of LSA translator, preempting the default behavior, which would only include it among the candidates to be elected as translator.

Note Even a forced translator might not translate all LSAs; translation depends on the contents of each LSA.

The figure below shows a network diagram in which OSPF Area 1 is defined as the stub area. The Enhanced Interior Gateway Routing Protocol (EIGRP) routes cannot be propagated into the OSPF domain because routing redistribution is not allowed in the stub area. However, once OSPF Area 1 is defined as an NSSA, an NSSA ASBR can inject the EIGRP routes into the OSPF NSSA by creating Type 7 LSAs.

Figure 201: OSPF NSSA



The redistributed routes from the RIP router will not be allowed into OSPF Area 1 because NSSA is an extension to the stub area. The stub area characteristics will still exist, including the exclusion of Type 5 LSAs.

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an ABR. In OSPF, an ABR will advertise networks in one area into another area. If the network numbers in an area are assigned in a way such that they are contiguous, you can configure the ABR to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

When routes from other protocols are redistributed into OSPF (as described in the module "Configuring IP Routing Protocol-Independent Features"), each route is advertised individually in an external LSA. However, you can configure the Cisco IOS software to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. Doing so helps decrease the size of the OSPF link-state database.

In OSPF, all areas must be connected to a backbone area. If there is a break in backbone continuity, or the backbone is purposefully partitioned, you can establish a virtual link. The two endpoints of a virtual link are ABRs. The virtual link must be configured in both routers. The configuration information in each router consists of the other virtual endpoint (the other ABR) and the nonbackbone area that the two routers have in common (called the transit area). Note that virtual links cannot be configured through stub areas.

You can force an ASBR to generate a default route into an OSPF routing domain. Whenever you specifically configure redistribution of routes into an OSPF routing domain, the router automatically becomes an ASBR. However, an ASBR does not, by default, generate a defaultroute into the OSPF routing domain.

You can configure OSPF to look up Domain Naming System (DNS) names for use in all OSPF show EXEC command displays. You can use this feature to more easily identify a router, because the router is displayed by name rather than by its router ID or neighbor ID.

OSPF uses the largest IP address configured on the interfaces as its router ID. If the interface associated with this IP address is ever brought down, or if the address is removed, the OSPF process must recalculate a new router ID and resend all its routing information out its interfaces.

If a loopback interface is configured with an IP address, the Cisco IOS software will use this IP address as its router ID, even if other interfaces have larger IP addresses. Because loopback interfaces never go down, greater stability in the routing table is achieved.

OSPF automatically prefers a loopback interface over any other kind, and it chooses the highest IP address among all loopback interfaces. If no loopback interfaces are present, the highest IP address in the router is chosen. You cannot tell OSPF to use any particular interface.

In Cisco IOS Release 10.3 and later releases, by default OSPF calculates the OSPF metric for an interface according to the bandwidth of the interface. For example, a 64-kbps link gets a metric of 1562, and a T1 link gets a metric of 64.

The OSPF metric is calculated as the ref-bw value divided by the bandwidth value, with the ref-bw value equal to 108 by default, and the bandwidth value determined by the bandwidth interface configuration command. The calculation gives FDDI a metric of 1. If you have multiple links with high bandwidth, you might want to specify a larger number to differentiate the cost on those links.

An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is an integer from 0 to 255. In general, the higher the value, the lower the trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored.

OSPF uses three different administrative distances: intra-area, interarea, and external. Routes within an area are intra-area; routes to another area are interarea; and routes from another routing domain learned via redistribution are external. The default distance for each type of route is 110.

Because simplex interfaces between two devices on an Ethernet represent only one network segment, for OSPF you must configure the sending interface to be a passive interface. This configuration prevents OSPF from sending hello packets for the sending interface. Both devices are able to see each other via the hello packet generated for the receiving interface.

You can configure the delay time between when OSPF receives a topology change and when it starts a shortest path first (SPF) calculation. You can also configure the hold time between two consecutive SPF calculations.

The OSPF on-demand circuit is an enhancement to the OSPF protocol that allows efficient operation over on-demand circuits such as ISDN, X.25 switched virtual circuits (SVCs), and dialup lines. This feature supports RFC 1793, Extending OSPF to Support Demand Circuits.

Prior to this feature, OSPF periodic hello and LSA updates would be exchanged between routers that connected the on-demand link, even when no changes occurred in the hello or LSA information.

With this feature, periodic hellos are suppressed and the periodic refreshes of LSAs are not flooded over the demand circuit. These packets bring up the link only when they are exchanged for the first time, or when a change occurs in the information they contain. This operation allows the underlying data link layer to be closed when the network topology is stable.

This feature is useful when you want to connect telecommuters or branch offices to an OSPF backbone at a central site. In this case, OSPF for on-demand circuits allows the benefits of OSPF over the entire domain, without excess connection costs. Periodic refreshes of hello updates, LSA updates, and other protocol overhead are prevented from enabling the on-demand circuit when there is no "real" data to send.

Overhead protocols such as hellos and LSAs are transferred over the on-demand circuit only upon initial setup and when they reflect a change in the topology. This means that critical changes to the topology that require new SPF calculations are sent in order to maintain network topology integrity. Periodic refreshes that do not include changes, however, are not sent across the link.

The OSPF LSA group pacing feature allows the router to group OSPF LSAs and pace the refreshing, checksumming, and aging functions. The group pacing results in more efficient use of the router.

The router groups OSPF LSAs and paces the refreshing, checksumming, and aging functions so that sudden increases in CPU usage and network resources are avoided. This feature is most beneficial to large OSPF networks.

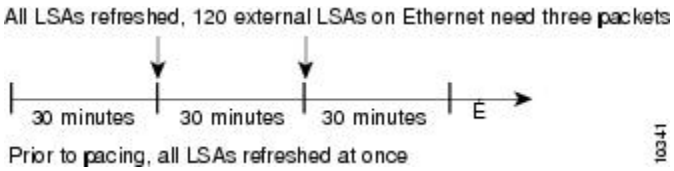
OSPF LSA group pacing is enabled by default. For typical customers, the default group pacing interval for refreshing, checksumming, and aging is appropriate and you need not configure this feature.

Original LSA Behavior

Each OSPF LSA has an age, which indicates whether the LSA is still valid. Once the LSA reaches the maximum age (1 hour), it is discarded. During the aging process, the originating router sends a refresh packet every 30 minutes to refresh the LSA. Refresh packets are sent to keep the LSA from expiring, whether there has been a change in the network topology or not. Checksumming is performed on all LSAs every 10 minutes. The router keeps track of LSAs that it generates and LSAs that it receives from other routers. The router refreshes LSAs that it generated; it ages the LSAs that it received from other routers.

Prior to the LSA group pacing feature, the Cisco software would perform refreshing on a single timer and checksumming and aging on another timer. In the case of refreshing, for example, the software would scan the whole database every 30 minutes, refreshing every LSA that the router generated, no matter how old it was. The figure below illustrates all the LSAs being refreshed at once. This process wasted CPU resources because only a small portion of the database needed to be refreshed. A large OSPF database (several thousand LSAs) could have thousands of LSAs with different ages. Refreshing on a single timer resulted in the age of all LSAs becoming synchronized, which resulted in much CPU processing at once. Furthermore, a large number of LSAs could cause a sudden increase of network traffic, consuming a large amount of network resources in a short time.

Figure 202: OSPF LSAs on a Single Timer Without Group Pacing



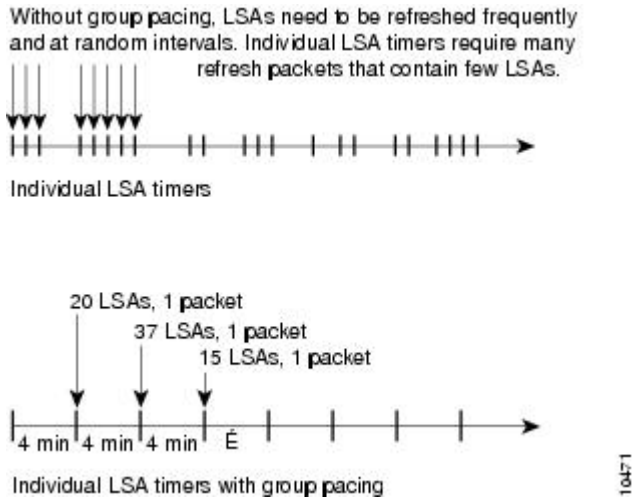
LSA Group Pacing with Multiple Timers

Configuring each LSA to have its own timer avoids excessive CPU processing and sudden network-traffic increase. To again use the example of refreshing, each LSA gets refreshed when it is 30 minutes old, independent of other LSAs. So the CPU is used only when necessary. However, LSAs being refreshed at frequent, random intervals would require many packets for the few refreshed LSAs that the router must send, which would be inefficient use of bandwidth.

Therefore, the router delays the LSA refresh function for an interval of time instead of performing it when the individual timers are reached. The accumulated LSAs constitute a group, which is then refreshed and sent out in one packet or more. Thus, the refresh packets are paced, as are the checksumming and aging. The pacing interval is configurable; it defaults to 4 minutes, which is randomized to further avoid synchronization.

The figure below illustrates the case of refresh packets. The first timeline illustrates individual LSA timers; the second timeline illustrates individual LSA timers with group pacing.

Figure 203: OSPF LSAs on Individual Timers with Group Pacing



The group pacing interval is inversely proportional to the number of LSAs that the router is refreshing, checksumming, and aging. For example, if you have approximately 10,000 LSAs, decreasing the pacing interval would benefit you. If you have a very small database (40 to 100 LSAs), increasing the pacing interval to 10 to 20 minutes might benefit you slightly.

The default value of pacing between LSA groups is 240 seconds (4 minutes). The range is from 10 seconds to 1800 seconds (30 minutes).

By default, OSPF floods new LSAs over all interfaces in the same area, except the interface on which the LSA arrives. Some redundancy is desirable, because it ensures robust flooding. However, too much redundancy can waste bandwidth and might destabilize the network due to excessive link and CPU usage in certain topologies. An example would be a fully meshed topology.

You can block OSPF flooding of LSAs in two ways, depending on the type of networks:

- On broadcast, nonbroadcast, and point-to-point networks, you can block flooding over specified OSPF interfaces.
- On point-to-multipoint networks, you can block flooding to a specified neighbor.

The growth of the Internet has increased the importance of scalability in IGPs such as OSPF. By design, OSPF requires LSAs to be refreshed as they expire after 3600 seconds. Some implementations have tried to improve the flooding by reducing the frequency to refresh from 30 minutes to about 50 minutes. This solution reduces the amount of refresh traffic but requires at least one refresh before the LSA expires. The OSPF flooding reduction solution works by reducing unnecessary refreshing and flooding of already known and unchanged information. To achieve this reduction, the LSAs are now flooded with the higher bit set. The LSAs are now set as “do not age.”

Cisco routers do not support LSA Type 6 Multicast OSPF (MOSPF), and they generate syslog messages if they receive such packets. If the router is receiving many MOSPF packets, you might want to configure the router to ignore the packets and thus prevent a large number of syslog messages.

The former OSPF implementation for sending update packets needed to be more efficient. Some update packets were getting lost in cases where the link was slow, a neighbor could not receive the updates quickly enough, or the router was out of buffer space. For example, packets might be dropped if either of the following topologies existed:

- A fast router was connected to a slower router over a point-to-point link.
- During flooding, several neighbors sent updates to a single router at the same time.

OSPF update packets are now automatically paced so they are not sent less than 33 milliseconds apart. Pacing is also added between resends to increase efficiency and minimize lost retransmissions. Also, you can display the LSAs waiting to be sent out an interface. The benefit of pacing is that OSPF update and retransmission packets are sent more efficiently. There are no configuration tasks for this feature; it occurs automatically.

You can display specific statistics such as the contents of IP routing tables, caches, and databases. Information provided can be used to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your device packets are taking through the network.

How to Configure OSPF

To configure OSPF, perform the tasks described in the following sections. The tasks in the "Enabling OSPF" section are required; the tasks in the remaining sections are optional, but might be required for your application. For information about the maximum number of interfaces, see the "Restrictions for OSPF" section.

Enabling OSPF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **network** *ip-address wildcard-mask* **area** *area-id*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing and enters router configuration mode.

	Command or Action	Purpose
Step 4	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Device(config-router)# network 192.168.129.16 0.0.0.3 area 20`	Defines an interface on which OSPF runs and defines the area ID for that interface.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring OSPF Interface Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf cost** *cost*
5. **ip ospf retransmit-interval** *seconds*
6. **ip ospf transmit-delay** *seconds*
7. **ip ospf priority** *number-value*
8. **ip ospf hello-interval** *seconds*
9. **ip ospf dead-interval** *seconds*
10. **ip ospf authentication-key** *key*
11. **ip ospf message-digest-key** *key-id* **md5** *key*
12. **ip ospf authentication** [**message-digest** | **null**]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface Gigabitethernet 0/0`	
Step 4	**ip ospf cost** *cost* **Example:** `Device(config-if)# ip ospf cost 65`	Explicitly specifies the cost of sending a packet on an OSPF interface.
Step 5	**ip ospf retransmit-interval** *seconds* **Example:** `Device(config-if)# ip ospf retransmit-interval 1`	Specifies the number of seconds between link-state advertisement (LSA) retransmissions for adjacencies belonging to an OSPF interface.
Step 6	**ip ospf transmit-delay** *seconds* **Example:** `Device(config-if)# ip ospf transmit-delay`	Sets the estimated number of seconds required to send a link-state update packet on an OSPF interface.
Step 7	**ip ospf priority** *number-value* **Example:** `Device(config-if)# ip ospf priority 1`	Sets priority to help determine the OSPF designated router for a network.
Step 8	**ip ospf hello-interval** *seconds* **Example:** `Device(config-if)# ip ospf hello-interval 1`	Specifies the length of time between the hello packets that the Cisco IOS software sends on an OSPF interface.
Step 9	**ip ospf dead-interval** *seconds* **Example:** `Device(config-if)# ip ospf dead-interval 1`	Sets the number of seconds that a device must wait before it declares a neighbor OSPF router down because it has not received a hello packet.
Step 10	**ip ospf authentication-key** *key* **Example:** `Device(config-if)# ip ospf authentication-key 1`	Assigns a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication.
Step 11	**ip ospf message-digest-key** *key-id* **md5** *key* **Example:** `Device(config-if)# ip ospf message-digest-key 1 md5 23456789`	Enables OSPF MD5 authentication. The values for the *key-id* and *key* arguments must match values specified for other neighbors on a network segment.
Step 12	**ip ospf authentication** [**message-digest** | **null**] **Example:**	Specifies the authentication type for an interface.

	Command or Action	Purpose
	`Device(config-if)# ip ospf authentication message-digest`	
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring OSPF over Different Physical Networks

Configuring OSPF for Point-to-Multipoint Broadcast Networks

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **ip ospf network point-to-multipoint**
4. **exit**
5. **router ospf** *process-id*
6. **neighbor** *ip-address* [**cost** *number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 3	**ip ospf network point-to-multipoint** **Example:** `Device#(config-if) ip ospf network point-to-multipoint`	Configures an interface as point-to-multipoint for broadcast media.
Step 4	**exit** **Example:** `Device#(config-if) exit`	Enters global configuration mode.

	Command or Action	Purpose
Step 5	**router ospf** *process-id* **Example:** `Device#(config) router ospf 109`	Configures an OSPF routing process and enters router configuration mode.
Step 6	**neighbor** *ip-address* [**cost** *number*] **Example:** `Device#(config-router) neighbor 192.168.3.4 cost 180`	Specifies a neighbor and assigns a cost to the neighbor. **Note** Repeat this step for each neighbor if you want to specify a cost. Otherwise, neighbors will assume the cost of the interface, based on the **ip ospf cost** interface configuration command.

Configuring OSPF for Nonbroadcast Networks

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **ip ospf network point-to-multipoint non-broadcast**
4. **exit**
5. **router ospf** *process-id*
6. **neighbor** *ip-address* [**cost** *number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 3	**ip ospf network point-to-multipoint non-broadcast** **Example:** `Device#(config-if) ip ospf network point-to-multipoint non-broadcast`	Configures an interface as point-to-multipoint for nonbroadcast media.
Step 4	**exit** **Example:** `Device#(config-if) exit`	Enters global configuration mode.

	Command or Action	Purpose
Step 5	**router ospf** *process-id* **Example:** `Device#(config) router ospf 109`	Configures an OSPF routing process and enters router configuration mode.
Step 6	**neighbor** *ip-address* [**cost** *number*] **Example:** `Device#(config-router) neighbor 192.168.3.4 cost 180`	Specifies a neighbor and assigns a cost to the neighbor. **Note** Repeat this step for each neighbor if you want to specify a cost. Otherwise, neighbors will assume the cost of the interface, based on the **ip ospf cost** interface configuration command.

Configuring OSPF Area Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **area** *area-id* **authentication**
5. **area** *area-id* **stub** [**no summary**]
6. **area** *area-id* **default-cost** *cost*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode.
Step 4	**area** *area-id* **authentication** **Example:**	Enables authentication for an OSPF area.

	Command or Action	Purpose
	`Device(config-router)# area 10.0.0.0 authentication`	
Step 5	**area** *area-id* **stub** [**no summary**] **Example:** `Device(config-router)# area 10.0.0.0 stub no-summary`	Defines an area to be a stub area.
Step 6	**area** *area-id* **default-cost** *cost* **Example:** `Device(config-router)# area 10.0.0.0 default-cost 1`	Specifies a cost for the default summary route that is sent into a stub area or not-so-stubby area (NSSA)
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring OSPFv2 NSSA

Configuring an OSPFv2 NSSA Area and Its Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*autonomous-system-number*] **[metric** {**metric-value** | **transparent**}] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] [**nssa-only**]
5. **network** *ip-address wildcard-mask* **area** *area-id*
6. **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate** [**metric**] [**metric-type**]] [**no-summary**] [**nssa-only**]
7. **summary-address** *prefix mask* [**not-advertise**] [**tag** *tag*] [**nssa-only**]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process. The range is from 1 to 65535.
Step 4	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*autonomous-system-number*] **[metric** {**metric-value** | **transparent**}] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] [**nssa-only**] **Example:** `Device(config-router)# redistribute rip subnets`	Redistributes routes from one routing domain to another routing domain. • In the example, Routing Information Protocol (RIP) subnets are redistributed into the OSPF domain.
Step 5	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Device(config-router)# network 192.168.129.11 0.0.0.255 area 1`	Defines the interfaces on which OSPF runs and the area ID for those interfaces.
Step 6	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate** [**metric**] [**metric-type**]] [**no-summary**] [**nssa-only**] **Example:** `Device(config-router)# area 1 nssa`	Configures a Not-So-Stubby Area (NSSA) area.
Step 7	**summary-address** *prefix mask* [**not-advertise**] [**tag** *tag*] [**nssa-only**] **Example:** `Device(config-router)# summary-address 10.1.0.0 255.255.0.0 not-advertise`	Controls the route summarization and filtering during the translation and limits the summary to NSSA areas.
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring an NSSA ABR as a Forced NSSA LSA Translator

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **area** *area-id* **nssa translate type7 always**
5. **area** *area-id* **nssa translate type7 suppress-fa**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 1`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process. The range is from 1 to 65535.
Step 4	**area** *area-id* **nssa translate type7 always** **Example:** `Device(config-router)# area 10 nssa translate type7 always`	Configures a Not-So-Stubby Area Area Border Router (NSSA ABR) device as a forced NSSA Link State Advertisement (LSA) translator. **Note** You can use the **always** keyword in the **area nssa translate** command to configure an NSSA ABR device as a forced NSSA LSA translator. This command can be used if RFC 3101 is disabled and RFC 1587 is used.
Step 5	**area** *area-id* **nssa translate type7 suppress-fa** **Example:** `Device(config-router)# area 10 nssa translate type7 suppress-fa`	Allows ABR to suppress the forwarding address in translated Type-5 LSA.
Step 6	**end** **Example:**	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router)# end`	

Disabling RFC 3101 Compatibility and Enabling RFC 1587 Compatibility

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **compatible rfc1587**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 1`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process. • Use **router ospf** *process-id* command to enable OSPFv2 routing.
Step 4	**compatible rfc1587** **Example:** `Device(config-router)# compatible rfc1587`	Enables the device to be RFC 1587 compliant.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring OSPF NSSA Parameters

Prerequisites

Evaluate the following considerations before you implement this feature:

- You can set a Type 7 default route that can be used to reach external destinations. When configured, the device generates a Type 7 default into the Not-So-Stubby Area (NSSA or the NSSA Area Border Router (ABR).
- Every device within the same area must agree that the area is NSSA; otherwise, the devices cannot communicate.

Configuring Route Summarization Between OSPF Areas

Configuring Route Summarization When Redistributing Routes into OSPF

SUMMARY STEPS

1. **summary-address** {*ip-address mask* | *prefix mask*} [**not-advertise**][**tag** *tag* [**nssa-only**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**summary-address** {*ip-address mask* | *prefix mask*} [**not-advertise**][**tag** *tag* [**nssa-only**] **Example:** `Device#(config-router) summary-address 10.1.0.0 255.255.0.0`	Specifies an address and mask that covers redistributed routes, so that only one summary route is advertised. • You can use the optional **not-advertise** keyword to filter out a set of routes.

Establishing Virtual Links

SUMMARY STEPS

1. **area** *area-id* **virtual-link** *router-id* [**authentication** [**message-digest** | **null**]] [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**authentication-key** *key* | **message-digest-key** *key-id* **md5** *key*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**area** *area-id* **virtual-link** *router-id* [**authentication** [**message-digest** | **null**]] [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**dead-interval** *seconds*] [**authentication-key** *key* | **message-digest-key** *key-id* **md5** *key*]	Establishes a virtual link.

	Command or Action	Purpose
	Example: `Device(config-router-af)# area 1 virtual-link 10.1.1.1 router1`	

Generating a Default Route

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing and enters router configuration mode.
Step 4	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*] **Example:** `Device(config-router)# default-information originate always`	Forces the ASBR to generate a default route into the OSPF routing domain. **Note** The **always** keyword includes the following exception when a route map is used. When a route map is used, the origination of the default route by OSPF is not bound to the existence of a default route in the routing table.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Lookup of DNS Names

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ospf name-lookup**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ospf name-lookup** **Example:** `Device# ip ospf name-lookup`	Enables OSPF routing and enters router configuration mode.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Forcing the Router ID Choice with a Loopback Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 2	**interface** *type number* **Example:** `Device(config)# interface loopback 0`	Creates a loopback interface and enters interface configuration mode.
Step 3	**ip address** *ip-address mask* **Example:** `Device#(config-if) ip address 192.108.1.27 255.255.255.0`	Assigns an IP address to this interface.

Controlling Default Metrics

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **auto-cost reference-bandwidth** *ref-bw*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device# router ospf 109`	Enables OSPF routing and enters router configuration mode.
Step 4	**auto-cost reference-bandwidth** *ref-bw* **Example:**	Differentiates high -bandwidth links.

	Command or Action	Purpose
	`Device(config-router)# auto-cost reference-bandwidth 101`	
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Changing the OSPF Administrative Distances

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **distance ospf** {**intra-area** | **inter-area** | **external**} *dist*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing and enters router configuration mode.
Step 4	**distance ospf** {**intra-area**	**inter-area**	**external**} *dist* **Example:** `Device(config-router)# distance ospf external 200`	Changes the OSPF distance values.
Step 5	**end** **Example:**	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router)# end`	

Configuring OSPF on Simplex Ethernet Interfaces

Command	Purpose
passive-interface *interface-type interface-number*	Suppresses the sending of hello packets through the specified interface.

Configuring Route Calculation Timers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **timers throttle spf** *spf-start spf-hold spf-max-wait*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing and enters router configuration mode.
Step 4	**timers throttle spf** *spf-start spf-hold spf-max-wait* **Example:** `Device(config-router)# timers throttle spf 5 1000 9000`	Configures route calculation timers.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring OSPF over On-Demand Circuits

SUMMARY STEPS

1. **router ospf** *process-id*
2. **interface** *type number*
3. **ip ospf demand-circuit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router ospf** *process-id*	Enables OSPF operation.
Step 2	**interface** *type number*	Enters interface configuration mode.
Step 3	**ip ospf demand-circuit**	Configures OSPF over an on-demand circuit.

What to do next

Note You can prevent an interface from accepting demand-circuit requests from other routers to by specifying the **ignore** keyword in the **ip ospf demand-circuit** command.

Prerequisites

Evaluate the following considerations before implementing the On-Demand Circuits feature:

- Because LSAs that include topology changes are flooded over an on-demand circuit, we recommend that you put demand circuits within OSPF stub areas or within NSSAs to isolate the demand circuits from as many topology changes as possible.
- Every router within a stub area or NSSA must have this feature loaded in order to take advantage of the on-demand circuit functionality. If this feature is deployed within a regular area, all other regular areas must also support this feature before the demand circuit functionality can take effect because Type 5 external LSAs are flooded throughout all areas.
- Hub-and-spoke network topologies that have a point-to-multipoint (P2MP) OSPF interface type on a hub might not revert to nondemand circuit mode when needed. You must simultaneously reconfigure OSPF on all interfaces on the P2MP segment when reverting them from demand circuit mode to nondemand circuit mode.

- Do not implement this feature on a broadcast-based network topology because the overhead protocols (such as hello and LSA packets) cannot be successfully suppressed, which means the link will remain up.
- Configuring the router for an OSPF on-demand circuit with an asynchronous interface is not a supported configuration. The supported configuration is to use dialer interfaces on both ends of the circuit. For more information, refer to Why OSPF Demand Circuit Keeps Bringing Up the Link .

Logging Neighbors Going Up or Down

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **log-adjacency-changes** [**detail**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing and enters router configuration mode.
Step 4	**log-adjacency-changes** [**detail**] **Example:** `Device(config-router)# log-adjacency-changes detail`	Changes the group pacing of LSAs. **Note** Configure the **log-adjacency-changes** command if you want to know about OSPF neighbors going up or down without turning on the **debug ip ospf adjacency** EXEC command because the **log-adjacency-changes** command provides a higher-level view of the peer relationship with less output. Configure the **log-adjacency-changes detail** command if you want to see messages for each state change.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Changing the LSA Group Pacing Interval

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **timers pacing lsa-group** *seconds*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing and enters router configuration mode.
Step 4	**timers pacing lsa-group** *seconds* **Example:** `Device(config-router)# timers pacing lsa-group 60`	Changes the group pacing of LSAs.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Blocking OSPF LSA Flooding

Command	Purpose
ip ospf database-filter all out	Blocks the flooding of OSPF LSA packets to the interface.

On point-to-multipoint networks, to block flooding of OSPF LSAs, use the following command in router configuration mode:

Command	Purpose
neighbor *ip-address* **database-filter all out**	Blocks the flooding of OSPF LSA packets to the specified neighbor.

Reducing LSA Flooding

Command	Purpose
ip ospf flood-reduction	Suppresses the unnecessary flooding of LSAs in stable topologies.

Ignoring MOSPF LSA Packets

Command	Purpose
ignore lsa mospf	Prevents the router from generating syslog messages when it receives MOSPF LSA packets.

Monitoring and Maintaining OSPF

Command	Purpose
show ip ospf [*process-id*]	Displays general information about OSPF routing processes.
show ip ospf border-routers	Displays the internal OSPF routing table entries to the ABR and ASBR.

Command	Purpose
	Displays lists of information related to the OSPF database.

Command	Purpose
show ip ospf [*process-id* [*area-id*]] **database** **show ip ospf** [*process-id* [*area-id*]] **database** [**database-summary**] **show ip ospf** [*process-id* [*area-id*]] **database** [**router**] [**self-originate**] **show ip ospf** [*process-id* [*area-id*]] **database** [**router**] [**adv-router** [*ip-address*]] **show ip ospf** [*process-id* [*area-id*]] **database** [**router**] [*link-state-id*] **show ip ospf** [*process-id* [*area-id*]] **database** [**network**] [*link-state-id*] **show ip ospf** [*process-id* [*area-id*]] **database** [**summary**] [*link-state-id*] **show ip ospf** [*process-id* [*area-id*]] **database** [**asbr-summary**] [*link-state-id*] **show ip ospf** [*process-id* [Router# *area-id*]] **database** [**external**] [*link-state-id*] **show ip ospf** [*process-id* [*area-id*]] **database** [**nssa-external**] [*link-state-id*] **show ip ospf** [*process-id* [*area-id*]] **database** [**opaque-link**] [*link-state-id*]	

Command	Purpose
show ip ospf [*process-id* [*area-id*]] **database** [**opaque-area**] [*link-state-id*] **show ip ospf** [*process-id* [*area-id*]] **database** [**opaque-as**] [*link-state-id*]	
show ip ospf flood-list interface *type*	Displays a list of LSAs waiting to be flooded over an interface (to observe OSPF packet pacing).
show ip ospf interface [*type number*]	Displays OSPF-related interface information.
show ip ospf neighbor [*interface-name*] [*neighbor-id*] **detail**	Displays OSPF neighbor information on a per-interface basis.
show ip ospf request-list [*neighbor*] [*interface*] [*interface-neighbor*]	Displays a list of all LSAs requested by a router.
show ip ospf retransmission-list [*neighbor*] [*interface*] [*interface-neighbor*]	Displays a list of all LSAs waiting to be re-sent.
show ip ospf [*process-id*] **summary-address**	Displays a list of all summary address redistribution information configured under an OSPF process.
show ip ospf virtual-links	Displays OSPF-related virtual links information.

To restart an OSPF process, use the following command in EXEC mode:

Command	Purpose
clear ip ospf [*pid*] {**process** | **redistribution** | **counters** [**neighbor** [**neighbor -** *interface*] [*neighbor-id*]]}	Clears redistribution based on the OSPF routing process ID. If the *pid* option is not specified, all OSPF processes are cleared.

Displaying OSPF Update Packet Pacing

SUMMARY STEPS

1. **show ip ospf flood-list** *interface-type interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip ospf flood-list** *interface-type interface-number* **Example:** `Device> show ip ospf flood-list ethernet 1`	Displays a list of OSPF LSAs waiting to be flooded over an interface.

Restrictions for OSPF

On systems with a large number of interfaces, it may be possible to configure OSPF such that the number of links advertised in the router LSA causes the link-state update packet to exceed the size of a "huge" Cisco buffer. To resolve this problem, reduce the number of OSPF links or increase the huge buffer size by entering the **buffers huge size** *size* command.

A link-state update packet containing a router LSA typically has a fixed overhead of 196 bytes, and an additional 12 bytes are required for each link description. With a huge buffer size of 18024 bytes, there can be a maximum of 1485 link descriptions.

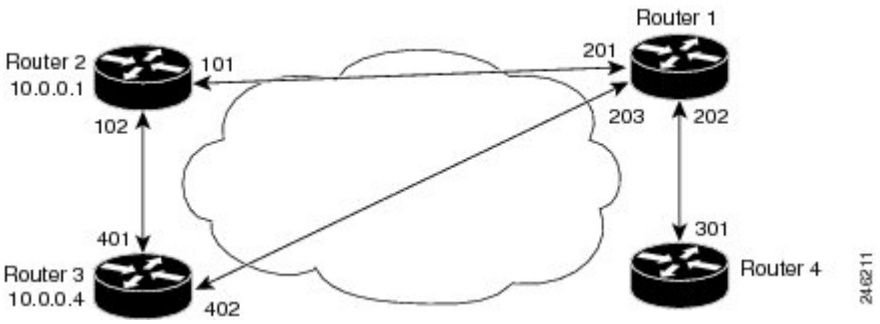
Because the maximum size of an IP packet is 65,535 bytes, there is still an upper bound on the number of links possible on a router.

Configuration Examples for OSPF

Example: OSPF Point-to-Multipoint

In the figure below, Router 1 uses data-link connection identifier (DLCI) 201 to communicate with Router 2, DLCI 202 to communicate with Router 4, and DLCI 203 to communicate with Router 3. Router 2 uses DLCI 101 to communicate with Router 1 and DLCI 102 to communicate with Router 3. Router 3 communicates with Router 2 (DLCI 401) and Router 1 (DLCI 402). Router 4 communicates with Router 1 (DLCI 301). Configuration examples follow the figure.

Figure 204: OSPF Point-to-Multipoint Example



Router 1 Configuration

```
hostname Router 1
!
```

```
interface serial 1
 ip address 10.0.0.2 255.0.0.0
 ip ospf network point-to-multipoint
 encapsulation frame-relay
 frame-relay map ip 10.0.0.1 201 broadcast
 frame-relay map ip 10.0.0.3 202 broadcast
 frame-relay map ip 10.0.0.4 203 broadcast
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
```

Router 2 Configuration

```
hostname Router 2
!
interface serial 0
 ip address 10.0.0.1 255.0.0.0
 ip ospf network point-to-multipoint
 encapsulation frame-relay
 frame-relay map ip 10.0.0.2 101 broadcast
 frame-relay map ip 10.0.0.4 102 broadcast
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
```

Router 3 Configuration

```
hostname Router 3
!
interface serial 3
 ip address 10.0.0.4 255.0.0.0
 ip ospf network point-to-multipoint
 encapsulation frame-relay
 clock rate 1000000
 frame-relay map ip 10.0.0.1 401 broadcast
 frame-relay map ip 10.0.0.2 402 broadcast
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
```

Router 4 Configuration

```
hostname Router 4
!
interface serial 2
 ip address 10.0.0.3 255.0.0.0
 ip ospf network point-to-multipoint
 encapsulation frame-relay
 clock rate 2000000
 frame-relay map ip 10.0.0.2 301 broadcast
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
```

Example: OSPF Point-to-Multipoint with Broadcast

The following example illustrates a point-to-multipoint network with broadcast:

```
interface Serial0
 ip address 10.0.1.1 255.255.255.0
 encapsulation frame-relay
 ip ospf cost 100
 ip ospf network point-to-multipoint
 frame-relay map ip 10.0.1.3 202 broadcast
 frame-relay map ip 10.0.1.4 203 broadcast
 frame-relay map ip 10.0.1.5 204 broadcast
 frame-relay local-dlci 200
!
router ospf 1
 network 10.0.1.0 0.0.0.255 area 0
 neighbor 10.0.1.5 cost 5
 neighbor 10.0.1.4 cost 10
```

The following example shows the configuration of the neighbor at 10.0.1.3:

```
interface serial 0
 ip address 10.0.1.3 255.255.255.0
 ip ospf network point-to-multipoint
 encapsulation frame-relay
 frame-relay local-dlci 301
 frame-relay map ip 10.0.1.1 300 broadcast
 no shutdown
!
 router ospf 1
 network 10.0.1.0 0.0.0.255 area 0
```

The output shown for neighbors in the first configuration is as follows:

```
Device# show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.1.1        1   FULL/  -        00:01:50    10.0.1.5        Serial0
172.16.1.4        1   FULL/  -        00:01:47    10.0.1.4        Serial0
172.16.1.8        1   FULL/  -        00:01:45    10.0.1.3        Serial0
```

The route information in the first configuration is as follows:

```
Device# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
Gateway of last resort is not set
C    1.0.0.0/8 is directly connected, Loopback0
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O       10.0.1.3/32 [110/100] via 10.0.1.3, 00:39:08, Serial0
C       10.0.1.0/24 is directly connected, Serial0
O       10.0.1.5/32 [110/5] via 10.0.1.5, 00:39:08, Serial0
O       10.0.1.4/32 [110/10] via 10.0.1.4, 00:39:08, Serial0
```

Example: OSPF Point-to-Multipoint with Nonbroadcast

The following example illustrates a point-to-multipoint network with nonbroadcast:

```
interface Serial0
 ip address 10.0.1.1 255.255.255.0
```

```
 ip ospf network point-to-multipoint non-broadcast
 encapsulation frame-relay
 no keepalive
 frame-relay local-dlci 200
 frame-relay map ip 10.0.1.3 202
 frame-relay map ip 10.0.1.4 203
 frame-relay map ip 10.0.1.5 204
 no shutdown
 !
 router ospf 1
  network 10.0.1.0 0.0.0.255 area 0
  neighbor 10.0.1.3 cost 5
  neighbor 10.0.1.4 cost 10
  neighbor 10.0.1.5 cost 15
```

The following example is the configuration for the router on the other side:

```
interface Serial9/2
 ip address 10.0.1.3 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint non-broadcast
 no ip mroute-cache
 no keepalive
 no fair-queue
 frame-relay local-dlci 301
 frame-relay map ip 10.0.1.1 300
 no shutdown
 !
 router ospf 1
  network 10.0.1.0 0.0.0.255 area 0
```

The output shown for neighbors in the first configuration is as follows:

```
Device# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.1.1        1   FULL/  -        00:01:52    10.0.1.5        Serial0
172.16.1.4        1   FULL/  -        00:01:52    10.0.1.4        Serial0
172.16.1.8        1   FULL/  -        00:01:52    10.0.1.3        Serial0
```

Example: Variable-Length Subnet Masks

OSPF, static routes, and IS-IS support variable-length subnet masks (VLSMs). With VLSMs, you can use different masks for the same network number on different interfaces, which allows you to conserve IP addresses and more efficiently use available address space.

In the following example, a 30-bit subnet mask is used, leaving two bits of address space reserved for serial-line host addresses. There is sufficient host address space for two host endpoints on a point-to-point serial link.

```
interface ethernet 0
 ip address 172.16.10.1 255.255.255.0
! 8 bits of host address space reserved for ethernets
interface serial 0
 ip address 172.16.20.1 255.255.255.252
! 2 bits of address space reserved for serial lines
! Router is configured for OSPF and assigned AS 107
router ospf 107
! Specifies network directly connected to the router
 network 172.16.0.0 0.0.255.255 area 0.0.0.0
```

Example: Configuring OSPF NSSA

In the following example, an Open Shortest Path First (OSPF) stub network is configured to include OSPF Area 0 and OSPF Area 1, using five devices. Device 3 is configured as the NSSA Autonomous System Border Router (ASBR). Device 2 configured to be the NSSA Area Border Router (ABR). OSPF Area 1 is defined as a Not-So-Stubby Area (NSSA).

Device 1

```
hostname Device1
!
interface Loopback1
 ip address 10.1.0.1 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.0.1 255.255.255.0
 ip ospf 1 area 0
 no cdp enable
!
interface Serial10/0
 description Device2 interface s11/0
 ip address 192.168.10.1 255.255.255.0
 ip ospf 1 area 1
 serial restart-delay 0
 no cdp enable
!
router ospf 1
 area 1 nssa
!
end
```

Device 2

```
hostname Device2
!
!
interface Loopback1
 ip address 10.1.0.2 255.255.255.255
!
interface Serial10/0
 description Device1 interface s11/0
 no ip address
 shutdown
 serial restart-delay 0
 no cdp enable
!
interface Serial11/0
 description Device1 interface s10/0
 ip address 192.168.10.2 255.255.255.0
 ip ospf 1 area 1
 serial restart-delay 0
 no cdp enable
!
interface Serial14/0
 description Device3 interface s13/0
 ip address 192.168.14.2 255.255.255.0
 ip ospf 1 area 1
 serial restart-delay 0
 no cdp enable
!
```

```
router ospf 1
 area 1 nssa
!
end
```

Device 3

```
hostname Device3
!
interface Loopback1
 ip address 10.1.0.3 255.255.255.255
!
interface Ethernet3/0
 ip address 192.168.3.3 255.255.255.0
 no cdp enable
!
interface Serial13/0
 description Device2 interface s14/0
 ip address 192.168.14.3 255.255.255.0
 ip ospf 1 area 1
 serial restart-delay 0
 no cdp enable
!
router ospf 1
 log-adjacency-changes
 area 1 nssa
 redistribute rip subnets
!
router rip
 version 2
 redistribute ospf 1 metric 15
 network 192.168.3.0
end
```

Device 4

```
hostname Device4
!
interface Loopback1
 ip address 10.1.0.4 255.255.255.255
!
interface Ethernet3/0
 ip address 192.168.3.4 255.255.255.0
 no cdp enable
!
interface Ethernet4/1
 ip address 192.168.41.4 255.255.255.0
!
router rip
 version 2
 network 192.168.3.0
 network 192.168.41.0
!
end
```

Device 5

```
hostname Device5
!
interface Loopback1
```

```
 ip address 10.1.0.5 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.0.10 255.255.255.0
 ip ospf 1 area 0
 no cdp enable
!
interface Ethernet1/1
 ip address 192.168.11.10 255.255.255.0
 ip ospf 1 area 0
!
router ospf 1
!
end
```

Example: OSPF NSSA Area with RFC 3101 Disabled and RFC 1587 Active

In the following example, the output for the **show ip ospf** and **show ip ospf database nssa** commands shows an Open Shortest Path First Not-So-Stubby Area (OSPF NSSA) area where RFC 3101 is disabled, RFC 1587 is active, and an NSSA Area Border Router (ABR) device is configured as a forced NSSA LSA translator. If RFC 3101 is disabled, the forced NSSA LSA translator remains inactive.

```
Device# show ip ospf

Routing Process "ospf 1" with ID 10.0.2.1
Start time: 00:00:25.512, Time elapsed: 00:01:02.200
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Supports NSSA (compatible with RFC 1587)
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
Router is not originating router-LSAs with maximum metric
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFs 10000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Incremental-SPF disabled
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 0 normal 0 stub 1 nssa
Number of areas transit capable is 0
External flood list length 0
IETF NSF helper support enabled
Cisco NSF helper support enabled
Reference bandwidth unit is 100 mbps
Area 1
Number of interfaces in this area is 1
It is a NSSA area
Configured to translate Type-7 LSAs, inactive (RFC3101 support
disabled)
Area has no authentication
SPF algorithm last executed 00:00:07.160 ago
SPF algorithm executed 3 times
Area ranges are
```

```
Number of LSA 3. Checksum Sum 0x0245F0
Number of opaque link LSA 0. Checksum Sum 0x000000
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

The table below describes the **show ip ospf** display fields and their descriptions.

Table 202: show ip ospf Field Descriptions

Field	Description
Supports NSSA (compatible with RFC 1587)	Specifies that RFC 1587 is active or that the OSPF NSSA area is RFC 1587 compatible.
Configured to translate Type-7 LSAs, inactive (RFC3101 support disabled)	Specifies that OSPF NSSA area has an ABR device configured to act as a forced translator of Type 7 LSAs. However, it is inactive because RFC 3101 is disabled

```
Device2# show ip ospf database nssa

Router Link States (Area 1)
LS age: 28
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 10.0.2.1
Advertising Router: 10.0.2.1
LS Seq Number: 80000004
Checksum: 0x5CA2
Length: 36
Area Border Router
AS Boundary Router
Unconditional NSSA translator
Number of Links: 1
Link connected to: a Stub Network
(Link ID) Network/subnet number: 192.0.2.5
(Link Data) Network Mask: 255.255.255.0
Number of MTID metrics: 0
TOS 0 Metrics: 10
```

The table below describes the **show ip ospf database nssa** display fields and their descriptions.

Table 203: show ip ospf database nssa Field Descriptions

Field	Description
Unconditional NSSA translator	Specifies that NSSA ASBR device is a forced NSSA LSA translator

Example: OSPF Routing and Route Redistribution

OSPF typically requires coordination among many internal routers, ABRs, and ASBRs. At a minimum, OSPF-based routers can be configured with all default parameter values, with no authentication, and with interfaces assigned to areas.

Three types of examples follow:

- The first is a simple configuration illustrating basic OSPF commands.

- The second example illustrates a configuration for an internal router, ABR, and ASBRs within a single, arbitrarily assigned, OSPF autonomous system.
- The third example illustrates a more complex configuration and the application of various tools available for controlling OSPF-based routing environments.

Example: Basic OSPF Configuration

The following example illustrates a simple OSPF configuration that enables OSPF routing process 9000, attaches Ethernet interface 0 to area 0.0.0.0, and redistributes RIP into OSPF and OSPF into RIP:

```
interface ethernet 0
 ip address 10.93.1.1 255.255.255.0
 ip ospf cost 1
!
interface ethernet 1
 ip address 10.94.1.1 255.255.255.0
!
router ospf 9000
 network 10.93.0.0 0.0.255.255 area 0.0.0.0
 redistribute rip metric 1 subnets
!
router rip
 network 10.94.0.0
 redistribute ospf 9000
 default-metric 1
```

Example: Basic OSPF Configuration for Internal Router ABR and ASBRs

The following example illustrates the assignment of four area IDs to four IP address ranges. In the example, OSPF routing process 109 is initialized, and four OSPF areas are defined: 10.9.50.0, 2, 3, and 0. Areas 10.9.50.0, 2, and 3 mask specific address ranges, and area 0 enables OSPF for *all other* networks.

```
router ospf 109
 network 192.168.10.0 0.0.0.255 area 10.9.50.0
 network 192.168.20.0 0.0.255.255 area 2
 network 192.168.30.0 0.0.0.255 area 3
 network 192.168.40.0 255.255.255.255 area 0
!
! Interface Ethernet0 is in area 10.9.50.0:
interface ethernet 0
 ip address 192.168.10.5 255.255.255.0
!
! Interface Ethernet1 is in area 2:
interface ethernet 1
 ip address 192.168.20.5 255.255.255.0
!
! Interface Ethernet2 is in area 2:
interface ethernet 2
 ip address 192.168.20.7 255.255.255.0
!
! Interface Ethernet3 is in area 3:
interface ethernet 3
 ip address 192.169.30.5 255.255.255.0
!
! Interface Ethernet4 is in area 0:
interface ethernet 4
 ip address 192.168.40.1 255.255.255.0
!
```

```
! Interface Ethernet5 is in area 0:
interface ethernet 5
 ip address 192.168.40.12 255.255.0.0
```

Each **network area** router configuration command is evaluated sequentially, so the order of these commands in the configuration is important. The Cisco software sequentially evaluates the address/wildcard-mask pair for each interface. See the **network area** command page in the *Cisco IOS IP Routing: OSPF Command Reference* for more information.

Consider the first **network area** command. Area ID 10.9.50.0 is configured for the interface on which subnet 192.168.10.0 is located. Assume that a match is determined for Ethernet interface 0. Ethernet interface 0 is attached to area 10.9.50.0 only.

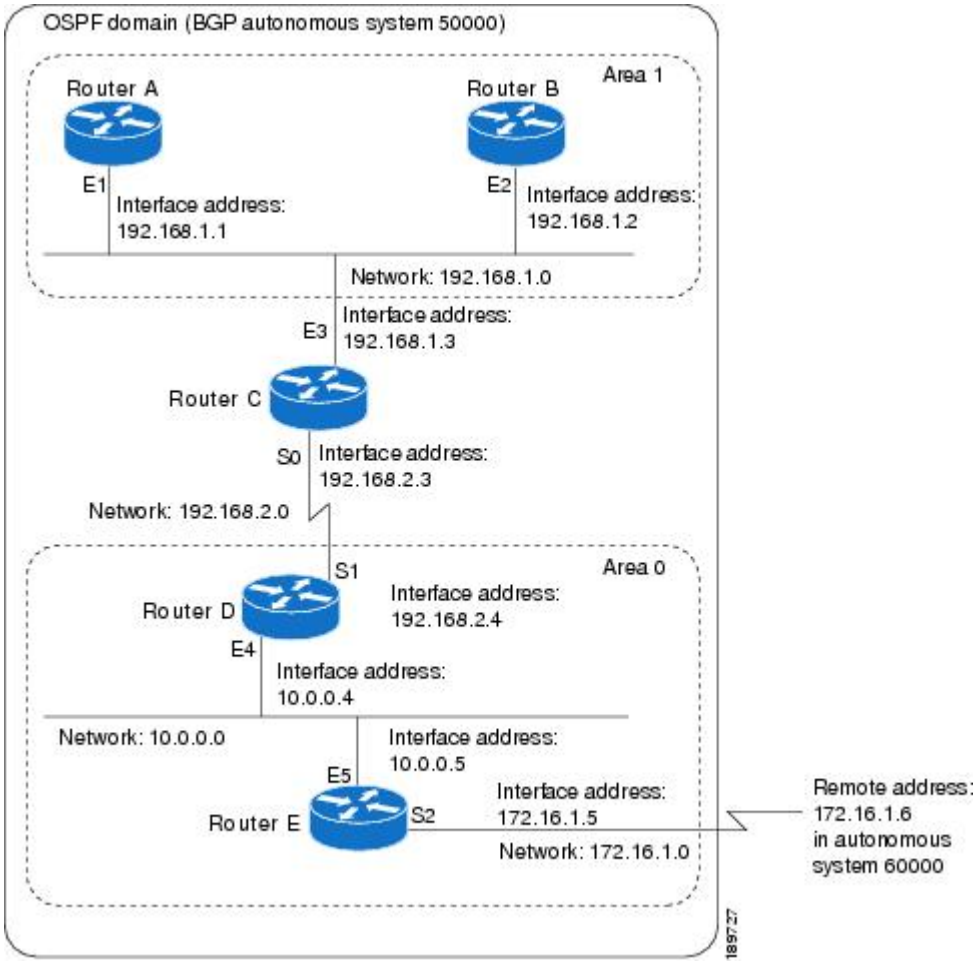
The second **network area** command is evaluated next. For area 2, the same process is then applied to all interfaces (except Ethernet interface 0). Assume that a match is determined for Ethernet interface 1. OSPF is then enabled for that interface, and Ethernet interface 1 is attached to area 2.

This process of attaching interfaces to OSPF areas continues for all **network area** commands. Note that the last **network area** command in this example is a special case. With this command, all available interfaces (not explicitly attached to another area) are attached to area 0.

Example: Complex Internal Router with ABR and ASBR

The following example outlines a configuration for several routers within a single OSPF autonomous system. The figure below provides a general network map that illustrates this sample configuration.

Figure 205: Sample OSPF Autonomous System Network Map



In this configuration, five routers are configured with OSPF:

- Router A and Router B are both internal routers within area 1.
- Router C is an OSPF ABR. Note that for Router C, Area 1 is assigned to E3 and area 0 is assigned to S0.
- Router D is an internal router in area 0 (backbone area). In this case, both **network** router configuration commands specify the same area (area 0, or the backbone area).
- Router E is an OSPF ASBR. Note that BGP routes are redistributed into OSPF and that these routes are advertised by OSPF.

Note You do not need to include definitions of all areas in an OSPF autonomous system in the configuration of all routers in the autonomous system. Only the *directly* connected areas must be defined. In the example that follows, routes in area 0 are learned by the routers in area 1 (Router A and Router B) when the ABR (Router C) injects summary LSAs into area 1.

The OSPF domain in BGP autonomous system 109 is connected to the outside world via the BGP link to the external peer at IP address 10.0.0.6. Sample configurations follow.

Following is the sample configuration for the general network map shown in the figure above.

Router A Configuration—Internal Router

```
interface ethernet 1
 ip address 192.168.1.1 255.255.255.0
router ospf 1
 network 192.168.0.0 0.0.255.255 area 1
```

Router B Configuration—Internal Router

```
interface ethernet 2
 ip address 192.168.1.2 255.255.255.0
router ospf 202
 network 192.168.0.0 0.0.255.255 area 1
```

Router C Configuration—ABR

```
interface ethernet 3
 ip address 192.168.1.3 255.255.255.0
interface serial 0
 ip address 192.168.2.3 255.255.255.0
router ospf 999
 network 192.168.1.0 0.0.0.255 area 1
 network 192.168.2.0 0.0.0.255 area 0
```

Router D Configuration—Internal Router

```
interface ethernet 4
 ip address 10.0.0.4 255.0.0.0
interface serial 1
 ip address 192.168.2.4 255.255.255.0
router ospf 50
 network 192.168.2.0 0.0.0.255 area 0
 network 10.0.0.0 0.255.255.255 area 0
```

Router E Configuration—ASBR

```
interface ethernet 5
 ip address 10.0.0.5 255.0.0.0
interface serial 2
 ip address 172.16.1.5 255.255.255.0
router ospf 65001
 network 10.0.0.0 0.255.255.255 area 0
 redistribute bgp 109 metric 1 metric-type 1
router bgp 109
 network 192.168.0.0
 network 10.0.0.0
 neighbor 172.16.1.6 remote-as 110
```

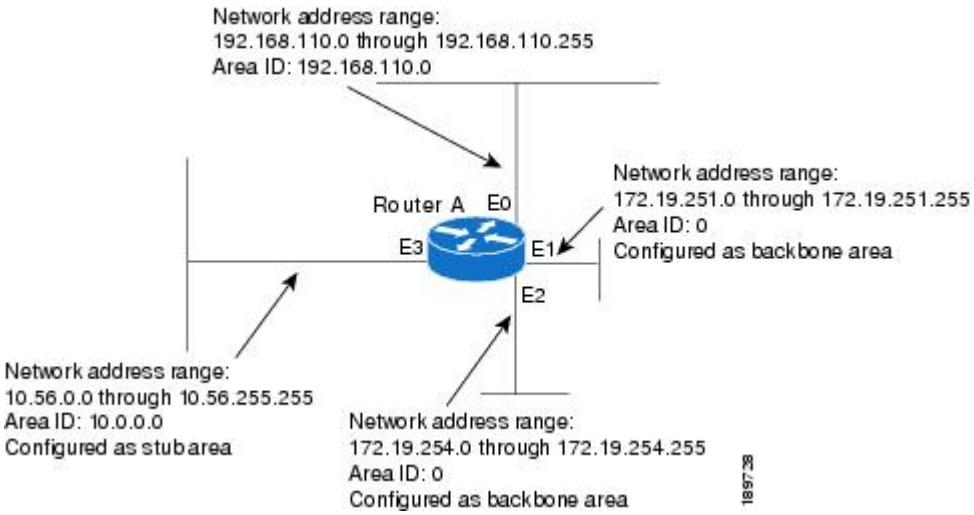
Example: Complex OSPF Configuration for ABR

The following sample configuration accomplishes several tasks in setting up an ABR. These tasks can be split into two general categories:

- Basic OSPF configuration
- Route redistribution

The specific tasks outlined in this configuration are detailed briefly in the following descriptions. The figure below illustrates the network address ranges and area assignments for the interfaces.

Figure 206: Interface and Area Specifications for OSPF Sample Configuration



The basic configuration tasks in this example are as follows:

- Configure address ranges for Ethernet interface 0 through Ethernet interface 3.
- Enable OSPF on each interface.
- Set up an OSPF authentication password for each area and network.
- Assign link-state metrics and other OSPF interface configuration options.
- Create a *stub area* with area ID 36.0.0.0. (Note that the **authentication** and **stub** options of the **area** router configuration command are specified with separate **area** command entries, but can be merged into a single **area** command.)
- Specify the backbone area (area 0).

Configuration tasks associated with redistribution are as follows:

- Redistribute IGRP and RIP into OSPF with various options set (including including **metric-type**, **metric**, **tag**, and **subnet**).
- Redistribute IGRP and OSPF into RIP.

The following is a sample OSPF configuration:

```
interface ethernet 0
```

```
 ip address 192.0.2.201 255.255.255.0
 ip ospf authentication-key abcdefgh
 ip ospf cost 10
!
interface ethernet 1
 ip address 172.19.251.202 255.255.255.0
 ip ospf authentication-key ijklmnop
 ip ospf cost 20
 ip ospf retransmit-interval 10
 ip ospf transmit-delay 2
 ip ospf priority 4
!
interface ethernet 2
 ip address 172.19.254.2 255.255.255.0
 ip ospf authentication-key abcdefgh
 ip ospf cost 10
!
interface ethernet 3
 ip address 10.56.0.0 255.255.0.0
 ip ospf authentication-key ijklmnop
 ip ospf cost 20
 ip ospf dead-interval 80
```

In the following configuration, OSPF is on network 172.16.0.0:

```
router ospf 201
 network 10.10.0.0 0.255.255.255 area 10.10.0.0
 network 192.42.110.0 0.0.0.255 area 192.42.110.0
 network 172.16.0.0 0.0.255.255 area 0
 area 0 authentication
 area 10.10.0.0 stub
 area 10.10.0.0 authentication
 area 10.10.0.0 default-cost 20
 area 192.42.110.0 authentication
 area 10.10.0.0 range 10.10.0.0 255.0.0.0
 area 192.42.110.0 range 192.42.110.0 255.255.255.0
 area 0 range 172.16.251.0 255.255.255.0
 area 0 range 172.16.254.0 255.255.255.0
 redistribute igrp 200 metric-type 2 metric 1 tag 200 subnets
 redistribute rip metric-type 2 metric 1 tag 200
```

In the following configuration, IGRP autonomous system 200 is on 192.0.2.1:

```
router igrp 200
 network 172.31.0.0
!
! RIP for 192.168.110
!
router rip
 network 192.168.110.0
 redistribute igrp 200 metric 1
 redistribute ospf 201 metric 1
```

Examples: Route Map

The examples in this section illustrate the use of redistribution, with and without route maps. Examples from the IP and Connectionless Network Service (CLNS) routing protocols are given.

The following example redistributes all OSPF routes into IGRP:

```
router igrp 109
 redistribute ospf 110
```

The following example redistributes RIP routes with a hop count equal to 1 into OSPF. These routes will be redistributed into OSPF as external LSAs with a metric of 5, a metric type of Type 1, and a tag equal to 1.

```
router ospf 109
 redistribute rip route-map rip-to-ospf
!
route-map rip-to-ospf permit
 match metric 1
 set metric 5
 set metric-type type1
 set tag 1
```

The following example redistributes OSPF learned routes with tag 7 as a RIP metric of 15:

```
router rip
 redistribute ospf 109 route-map 5
!
route-map 5 permit
 match tag 7
 set metric 15
```

The following example redistributes OSPF intra-area and interarea routes with next-hop routers on serial interface 0 into BGP with an INTER_AS metric of 5:

```
router bgp 109
 redistribute ospf 109 route-map 10
!
route-map 10 permit
 match route-type internal
 match interface serial 0
 set metric 5
```

The following example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first type is OSPF external IP routes with tag 5; these routes are inserted into Level 2 IS-IS link state packets (LSPs) with a metric of 5. The second type is ISO-IGRP derived CLNS prefix routes that match CLNS access list 2000; these routes will be redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
router isis
 redistribute ospf 109 route-map 2
 redistribute iso-igrp nsfnet route-map 3
!
route-map 2 permit
 match route-type external
 match tag 5
 set metric 5
 set level level-2
!
route-map 3 permit
 match address 2000
 set metric 30
```

With the following configuration, OSPF external routes with tags 1, 2, 3, and 5 are redistributed into RIP with metrics of 1, 1, 5, and 5, respectively. The OSPF routes with a tag of 4 are not redistributed.

```
router rip
```

```
 redistribute ospf 109 route-map 1
!
route-map 1 permit
 match tag 1 2
 set metric 1
!
route-map 1 permit
 match tag 3
 set metric 5
!
route-map 1 deny
 match tag 4
!
route map 1 permit
 match tag 5
 set metric 5
```

In the following configuration, a RIP-learned route for network 192.168.0.0 and an ISO-IGRP-learned route with prefix 49.0001.0002 are redistributed into an IS-IS Level 2 LSP with a metric of 5:

```
router isis
 redistribute rip route-map 1
 redistribute iso-igrp remote route-map 1
!
route-map 1 permit
 match ip address 1
 match clns address 2
 set metric 5
 set level level-2
!
 access-list 1 permit 192.168.0.0 0.0.255.255
 clns filter-set 2 permit 49.0001.0002...
```

The following configuration example illustrates how a route map is referenced by the **default-information** router configuration command. This type of reference is called *conditional default origination*. OSPF will originate the default route (network 0.0.0.0) with a Type 2 metric of 5 if 172.16.0.0 is in the routing table.

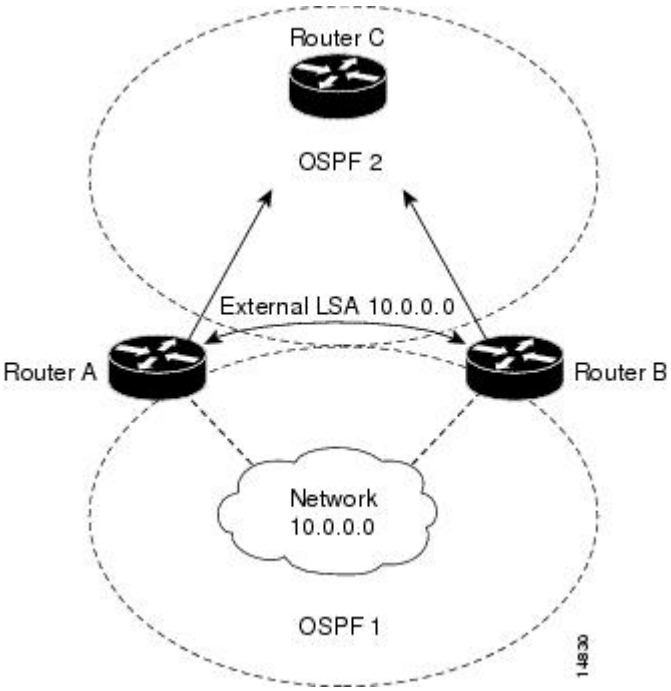
Note Only routes external to the OSPF process can be used for tracking, such as non-OSPF routes or OSPF routes from a separate OSPF process.

```
route-map ospf-default permit
 match ip address 1
 set metric 5
 set metric-type type-2
!
 access-list 1 permit 172.16.0.0 0.0.255.255
!
router ospf 109
 default-information originate route-map ospf-default
```

Example: Changing the OSPF Administrative Distances

The following configuration changes the external distance to 200, making it less trustworthy. The figure below illustrates the example.

Figure 207: OSPF Administrative Distance



Router A Configuration

```
router ospf 1
 redistribute ospf 2 subnet
 distance ospf external 200
!
router ospf 2
 redistribute ospf 1 subnet
 distance ospf external 200
```

Router B Configuration

```
router ospf 1
 redistribute ospf 2 subnet
 distance ospf external 200
!
router ospf 2
 redistribute ospf 1 subnet
 distance ospf external 200
```

Example: OSPF over On-Demand Routing

The following configuration allows OSPF over an on-demand circuit, as shown in the figure below. Note that the on-demand circuit is defined on one side only (BRI 0 on Router A); it is not required to be configured on both sides.

Figure 208: OSPF over On-Demand Circuit

Router A Configuration

```
username RouterB password 7 060C1A2F47
isdn switch-type basic-5ess
ip routing
!
interface TokenRing0
 ip address 192.168.50.5 255.255.255.0
 no shutdown
!
interface BRI0
 no cdp enable
 description connected PBX 1485
 ip address 192.168.45.30 255.255.255.0
 encapsulation ppp
 ip ospf demand-circuit
 dialer map ip 192.0.2.6 name RouterB broadcast 61484
 dialer-group 1
 ppp authentication chap
 no shutdown
!
router ospf 100
 network 192.168.45.0 0.0.0.255 area 0
 network 192.168.45.50 0.0.0.255 area 0
!
dialer-list 1 protocol ip permit
```

Router B Configuration

```
username RouterA password 7 04511E0804
isdn switch-type basic-5ess
ip routing
!
interface Ethernet0
 ip address 192.168.50.16 255.255.255.0
 no shutdown
!
interface BRI0
 no cdp enable
 description connected PBX 1484
 ip address  192.168.45.17 255.255.255.0
 encapsulation ppp
 dialer map ip 192.168.45.19 name RouterA broadcast 61485
 dialer-group 1
 ppp authentication chap
 no shutdown
!
router ospf 100
 network 192.168.45.0 0.0.0.255 area 0
 network 192.168.45.50 0.0.0.255 area 0
!
dialer-list 1 protocol ip permit
```

Example: LSA Group Pacing

The following example changes the OSPF pacing between LSA groups to 60 seconds:

```
router ospf
 timers pacing lsa-group 60
```

Example: Blocking OSPF LSA Flooding

The following example prevents flooding of OSPF LSAs to broadcast, nonbroadcast, or point-to-point networks reachable through Ethernet interface 0:

```
interface ethernet 0
 ip ospf database-filter all out
```

The following example prevents flooding of OSPF LSAs to point-to-multipoint networks to the neighbor at IP address 10.10.10.45:

```
router ospf 109
 neighbor 10.10.10.45 database-filter all out
```

Example: Ignoring MOSPF LSA Packets

The following example configures the router to suppress the sending of syslog messages when it receives MOSPF packets:

```
router ospf 109
 ignore lsa mospf
```

Additional References for OSPF Not-So-Stubby Areas (NSSA)

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
Protocol-independent features that work with OSPF	"Configuring IP Routing Protocol-Independent Features" module in *IP Routing: Protocol-Independent Configuration Guide*

RFCs

RFC	Title
RFC 1587	The OSPF NSSA Option , March 1994
RFC 3101	The OSPF NSSA Option January 2003

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring OSPF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 194

IPv6 Routing: OSPFv3

Open Shortest Path First version 3 (OSPFv3) is an IPv4 and IPv6 link-state routing protocol that supports IPv6 and IPv4 unicast address families (AFs).

Prerequisites for IPv6 Routing: OSPFv3

- Complete the OSPFv3 network strategy and planning for your IPv6 network. For example, you must decide whether multiple areas are required.
- Enable IPv6 unicast routing.
- Enable IPv6 on the interface.

Restrictions for IPv6 Routing: OSPFv3

When running a dual-stack IP network with OSPF version 2 for IPv4 and OSPFv3, be careful when changing the defaults for commands used to enable OSPFv3. Changing these defaults may affect your OSPFv3 network, possibly adversely.

Information About IPv6 Routing: OSPFv3

How OSPFv3 Works

OSPFv3 is a routing protocol for IPv4 and IPv6. It is a link-state protocol, as opposed to a distance-vector protocol. Think of a link as being an interface on a networking device. A link-state protocol makes its routing

decisions based on the states of the links that connect source and destination machines. The state of a link is a description of that interface and its relationship to its neighboring networking devices. The interface information includes the IPv6 prefix of the interface, the network mask, the type of network it is connected to, the devices connected to that network, and so on. This information is propagated in various type of link-state advertisements (LSAs).

A device's collection of LSA data is stored in a link-state database. The contents of the database, when subjected to the Dijkstra algorithm, result in the creation of the OSPF routing table. The difference between the database and the routing table is that the database contains a complete collection of raw data; the routing table contains a list of shortest paths to known destinations via specific device interface ports.

OSPFv3, which is described in RFC 5340, supports IPv6 and IPv4 unicast AFs.

Comparison of OSPFv3 and OSPF Version 2

Much of OSPF version 3 is the same as in OSPF version 2. OSPFv3, which is described in RFC 5340, expands on OSPF version 2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.

In OSPFv3, a routing process does not need to be explicitly created. Enabling OSPFv3 on an interface will cause a routing process, and its associated configuration, to be created.

In OSPFv3, each interface must be enabled using commands in interface configuration mode. This feature is different from OSPF version 2, in which interfaces are indirectly enabled using the device configuration mode.

When using a nonbroadcast multiaccess (NBMA) interface in OSPFv3, you must manually configure the device with the list of neighbors. Neighboring devices are identified by their device ID.

In IPv6, you can configure many address prefixes on an interface. In OSPFv3, all address prefixes on an interface are included by default. You cannot select some address prefixes to be imported into OSPFv3; either all address prefixes on an interface are imported, or no address prefixes on an interface are imported.

Unlike OSPF version 2, multiple instances of OSPFv3 can be run on a link.

OSPF automatically prefers a loopback interface over any other kind, and it chooses the highest IP address among all loopback interfaces. If no loopback interfaces are present, the highest IP address in the device is chosen. You cannot tell OSPF to use any particular interface.

LSA Types for OSPFv3

The following list describes LSA types, each of which has a different purpose:

- Device LSAs (Type 1)—Describes the link state and costs of a device's links to the area. These LSAs are flooded within an area only. The LSA indicates if the device is an Area Border Router (ABR) or Autonomous System Boundary Router (ASBR), and if it is one end of a virtual link. Type 1 LSAs are also used to advertise stub networks. In OSPFv3, these LSAs have no address information and are network-protocol-independent. In OSPFv3, device interface information may be spread across multiple device LSAs. Receivers must concatenate all device LSAs originated by a given device when running the SPF calculation.
- Network LSAs (Type 2)—Describes the link-state and cost information for all devices attached to the network. This LSA is an aggregation of all the link-state and cost information in the network. Only a designated device tracks this information and can generate a network LSA. In OSPFv3, network LSAs have no address information and are network-protocol-independent.

- Interarea-prefix LSAs for ABRs (Type 3)—Advertises internal networks to devices in other areas (interarea routes). Type 3 LSAs may represent a single network or a set of networks summarized into one advertisement. Only ABRs generate summary LSAs. In OSPFv3, addresses for these LSAs are expressed as *prefix*, *prefix length* instead of *address*, *mask*. The default route is expressed as a prefix with length 0.
- Interarea-device LSAs for ASBRs (Type 4)—Advertises the location of an ASBR. Devices that are trying to reach an external network use these advertisements to determine the best path to the next hop. Type 4 LSAs are generated by ABRs on behalf of ASBRs.
- Autonomous system external LSAs (Type 5)—Redistributes routes from another autonomous system, usually from a different routing protocol into OSPFv3. In OSPFv3, addresses for these LSAs are expressed as *prefix*, *prefix length* instead of *address*, *mask*. The default route is expressed as a prefix with length 0.
- Link LSAs (Type 8)—Have local-link flooding scope and are never flooded beyond the link with which they are associated. Link LSAs provide the link-local address of the device to all other devices attached to the link, inform other devices attached to the link of a list of prefixes to associate with the link, and allow the device to assert a collection of Options bits to associate with the network LSA that will be originated for the link.
- Intra-Area-Prefix LSAs (Type 9)—A device can originate multiple intra-area-prefix LSAs for each device or transit network, each with a unique link-state ID. The link-state ID for each intra-area-prefix LSA describes its association to either the device LSA or the network LSA and contains prefixes for stub and transit networks.

An address prefix occurs in almost all newly defined LSAs. The prefix is represented by three fields: PrefixLength, PrefixOptions, and Address Prefix. In OSPFv3, addresses for these LSAs are expressed as *prefix*, *prefix length* instead of *address*, *mask*. The default route is expressed as a prefix with length 0. Type 3 and Type 9 LSAs carry all prefix (subnet) information that, in OSPFv2, is included in device LSAs and network LSAs. The Options field in certain LSAs (device LSAs, network LSAs, interarea-device LSAs, and link LSAs) has been expanded to 24 bits to provide support for OSPFv3.

In OSPFv3, the sole function of the link-state ID in interarea-prefix LSAs, interarea-device LSAs, and autonomous-system external LSAs is to identify individual pieces of the link-state database. All addresses or device IDs that are expressed by the link-state ID in OSPF version 2 are carried in the body of the LSA in OSPFv3.

The link-state ID in network LSAs and link LSAs is always the interface ID of the originating device on the link being described. For this reason, network LSAs and link LSAs are now the only LSAs whose size cannot be limited. A network LSA must list all devices connected to the link, and a link LSA must list all of the address prefixes of a device on the link.

Load Balancing in OSPFv3

When a device learns multiple routes to a specific network via multiple routing processes (or routing protocols), it installs the route with the lowest administrative distance in the routing table. Sometimes the device must select a route from among many learned via the same routing process with the same administrative distance. In this case, the device chooses the path with the lowest cost (or metric) to the destination. Each routing process calculates its cost differently and the costs may need to be manipulated in order to achieve load balancing.

OSPFv3 performs load balancing automatically in the following way. If OSPFv3 finds that it can reach a destination through more than one interface and each path has the same cost, it installs each path in the routing

table. The only restriction on the number of paths to the same destination is controlled by the **maximum-paths** command. The default maximum paths is 16, and the range is from 1 to 64.

Addresses Imported into OSPFv3

When importing the set of addresses specified on an interface on which OSPFv3 is running into OSPFv3, you cannot select specific addresses to be imported. Either all addresses are imported, or no addresses are imported.

OSPFv3 Customization

You can customize OSPFv3 for your network, but you likely will not need to do so. The defaults for OSPFv3 are set to meet the requirements of most customers and features. If you must change the defaults, refer to the IPv6 command reference to find the appropriate syntax.

Caution Be careful when changing the defaults. Changing defaults will affect your OSPFv3 network, possibly adversely.

Force SPF in OSPFv3

When the **process** keyword is used with the **clear ipv6 ospf** command, the OSPFv3 database is cleared and repopulated, and then the SPF algorithm is performed. When the **force-spf** keyword is used with the **clear ipv6 ospf** command, the OSPFv3 database is not cleared before the SPF algorithm is performed.

How to Configure Load Balancing in OSPFv3

Configuring the OSPFv3 Device Process

Once you have completed step 3 and entered OSPFv3 router configuration mode, you can perform any of the subsequent steps in this task as needed to configure OSPFv3 Device configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **area** *area-ID* [**default-cost** | **nssa** | **stub**]
5. **auto-cost reference-bandwidth** *Mbps*
6. **default** {**area** *area-ID* [**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*]
7. **ignore lsa mospf**
8. **interface-id snmp-if-index**
9. **log-adjacency-changes** [**detail**]
10. **passive-interface** [**default** | *interface-type interface-number*]
11. **queue-depth** {**hello** | **update**} {*queue-size* | **unlimited**}

12. **router-id** *router-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enters router configuration mode for the IPv4 or IPv6 address family.
Step 4	**area** *area-ID* [**default-cost** | **nssa** | **stub**] **Example:** `Device(config-router)# area 1`	Configures the OSPFv3 area.
Step 5	**auto-cost reference-bandwidth** *Mbps* **Example:** `Device(config-router)# auto-cost reference-bandwidth 1000`	Controls the reference value OSPFv3 uses when calculating metrics for interfaces in an IPv4 OSPFv3 process.
Step 6	**default** {**area** *area-ID* [**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*] **Example:** `Device(config-router)# default area 1`	Returns an OSPFv3 parameter to its default value.
Step 7	**ignore lsa mospf** **Example:** `Device(config-router)# ignore lsa mospf`	Suppresses the sending of syslog messages when the device receives LSA Type 6 multicast OSPFv3 packets, which are unsupported.

	Command or Action	Purpose
Step 8	**interface-id snmp-if-index** **Example:** `Device(config-router)# interface-id snmp-if-index`	Configures OSPFv3 interfaces with Simple Network Management Protocol (SNMP) MIB-II interface Index (ifIndex) identification numbers in IPv4 and IPv6.
Step 9	**log-adjacency-changes** [**detail**] **Example:** `Device(config-router)# log-adjacency-changes`	Configures the device to send a syslog message when an OSPFv3 neighbor goes up or down.
Step 10	**passive-interface** [**default** | *interface-type interface-number*] **Example:** `Device(config-router)# passive-interface default`	Suppresses sending routing updates on an interface when an IPv4 OSPFv3 process is used.
Step 11	**queue-depth** {**hello** | **update**} {*queue-size* | **unlimited**} **Example:** `Device(config-router)# queue-depth update 1500`	Configures the number of incoming packets that the IPv4 OSPFv3 process can keep in its queue.
Step 12	**router-id** *router-id* **Example:** `Device(config-router)# router-id 10.1.1.1`	Enter this command to use a fixed router ID.

Forcing an SPF Calculation

SUMMARY STEPS

1. **enable**
2. **clear ospfv3** [*process-id*] **force-spf**
3. **clear ospfv3** [*process-id*] **process**
4. **clear ospfv3** [*process-id*] **redistribution**
5. **clear ipv6 ospf** [*process-id*] {**process** | **force-spf** | **redistribution**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**clear ospfv3** [*process-id*] **force-spf** **Example:** `Device# clear ospfv3 1 force-spf`	Runs SPF calculations for an OSPFv3 process. • If the **clear ospfv3 force-spf** command is configured, it overwrites the **clear ipv6 ospf** configuration. • Once the **clear ospfv3 force-spf** command has been used, the **clear ipv6 ospf** command cannot be used.
Step 3	**clear ospfv3** [*process-id*] **process** **Example:** `Device# clear ospfv3 2 process`	Resets an OSPFv3 process. • If the **clear ospfv3 force-spf** command is configured, it overwrites the **clear ipv6 ospf** configuration. • Once the **clear ospfv3 force-spf** command has been used, the **clear ipv6 ospf** command cannot be used.
Step 4	**clear ospfv3** [*process-id*] **redistribution** **Example:** `Device# clear ospfv3 redistribution`	Clears OSPFv3 route redistribution. • If the **clear ospfv3 force-spf** command is configured, it overwrites the **clear ipv6 ospf** configuration. • Once the **clear ospfv3 force-spf** command has been used, the **clear ipv6 ospf** command cannot be used.
Step 5	**clear ipv6 ospf** [*process-id*] {**process** | **force-spf** | **redistribution**} **Example:** `Device# clear ipv6 ospf force-spf`	Clears the OSPFv3 state based on the OSPFv3 routing process ID, and forces the start of the SPF algorithm. • If the **clear ospfv3 force-spf** command is configured, it overwrites the **clear ipv6 ospf** configuration. • Once the **clear ospfv3 force-spf** command has been used, the **clear ipv6 ospf** command cannot be used.

Verifying OSPFv3 Configuration and Operation

This task is optional, and the commands can be entered in any order, as needed.

SUMMARY STEPS

1. **enable**
2. **show ospfv3** [*process-id*] [*address-family*] **border-routers**
3. **show ospfv3** [*process-id* [*area-id*]] [*address-family*] **database** [**database-summary** | **internal** | **external** [*ipv6-prefix*] [*link-state-id*] | **grace** | **inter-area prefix** [*ipv6-prefix* | *link-state-id*] | **inter-area router** [*destination-router-id* | *link-state-id*] | **link** [**interface** *interface-name* | *link-state-id*] | **network** [*link-state-id*] | **nssa-external** [*ipv6-prefix*] [*link-state-id*] | **prefix** [**ref-lsa** {**router** | **network**} | *link-state-id*] | **promiscuous** | **router** [*link-state-id*] | **unknown** [{**area** | **as** | **link**} [*link-state-id*]] [**adv-router** *router-id*] [**self-originate**]
4. **show ospfv3** [*process-id*] [*address-family*] **events** [**generic** | **interface** | **lsa** | **neighbor** | **reverse** | **rib** | **spf**]
5. **show ospfv3** [*process-id*] [*area-id*] [*address-family*] **flood-list** *interface-type interface-number*

6. **show ospfv3** [*process-id*] [*address-family*] **graceful-restart**
7. **show ospfv3** [*process-id*] [*area-id*] [*address-family*] **interface** [*type number*] [**brief**]
8. **show ospfv3** [*process-id*] [*area-id*] [*address-family*] **neighbor** [*interface-type interface-number*] [*neighbor-id*] [**detail**]
9. **show ospfv3** [*process-id*] [*area-id*] [*address-family*] **request-list**[*neighbor*] [*interface*] [*interface-neighbor*]
10. **show ospfv3** [*process-id*] [*area-id*] [*address-family*] **retransmission-list** [*neighbor*] [*interface*] [*interface-neighbor*]
11. **show ospfv3** [*process-id*] [*address-family*] **statistic** [**detail**]
12. **show ospfv3** [*process-id*] [*address-family*] **summary-prefix**
13. **show ospfv3** [*process-id*] [*address-family*] **timers rate-limit**
14. **show ospfv3** [*process-id*] [*address-family*] **traffic**[*interface-type interface-number*]
15. **show ospfv3** [*process-id*] [*address-family*] **virtual-links**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ospfv3** [*process-id*] [*address-family*] **border-routers** **Example:** `Device# show ospfv3 border-routers`	Displays the internal OSPFv3 routing table entries to an ABR and ASBR.
Step 3	**show ospfv3** [*process-id* [*area-id*]] [*address-family*] **database** [**database-summary** | **internal** | **external** [*ipv6-prefix*] [*link-state-id*] | **grace** | **inter-area prefix** [*ipv6-prefix* | *link-state-id*] | **inter-area router** [*destination-router-id* | *link-state-id*] | **link** [**interface** *interface-name* | *link-state-id*] | **network** [*link-state-id*] | **nssa-external** [*ipv6-prefix*] [*link-state-id*] | **prefix** [**ref-lsa** {**router** | **network**} | *link-state-id*] | **promiscuous** | **router** [*link-state-id*] | **unknown** [{**area** | **as** | **link**} [*link-state-id*]] [**adv-router** *router-id*] [**self-originate**] **Example:** `Device# show ospfv3 database`	Displays lists of information related to the OSPFv3 database for a specific device.
Step 4	**show ospfv3** [*process-id*] [*address-family*] **events** [**generic** | **interface** | **lsa** | **neighbor** | **reverse** | **rib** | **spf**] **Example:** `Device# show ospfv3 events`	Displays detailed information about OSPFv3 events.

	Command or Action	Purpose
Step 5	**show ospfv3** [*process-id*] [*area-id*] [*address-family*] **flood-list** *interface-type interface-number* **Example:** `Device# show ospfv3 flood-list`	Displays a list of OSPFv3 LSAs waiting to be flooded over an interface.
Step 6	**show ospfv3** [*process-id*] [*address-family*] **graceful-restart** **Example:** `Device# show ospfv3 graceful-restart`	Displays OSPFv3 graceful restart information.
Step 7	**show ospfv3** [*process-id*] [*area-id*] [*address-family*] **interface** [*type number*] [**brief**] **Example:** `Device# show ospfv3 interface`	Displays OSPFv3-related interface information.
Step 8	**show ospfv3** [*process-id*] [*area-id*] [*address-family*] **neighbor** [*interface-type interface-number*] [*neighbor-id*] [**detail**] **Example:** `Device# show ospfv3 neighbor`	Displays OSPFv3 neighbor information on a per-interface basis.
Step 9	**show ospfv3** [*process-id*] [*area-id*] [*address-family*] **request-list**[*neighbor*] [*interface*] [*interface-neighbor*] **Example:** `Device# show ospfv3 request-list`	Displays a list of all LSAs requested by a device.
Step 10	**show ospfv3** [*process-id*] [*area-id*] [*address-family*] **retransmission-list** [*neighbor*] [*interface*] [*interface-neighbor*] **Example:** `Device# show ospfv3 retransmission-list`	Displays a list of all LSAs waiting to be re-sent.
Step 11	**show ospfv3** [*process-id*] [*address-family*] **statistic** [**detail**] **Example:** `Device# show ospfv3 statistic`	Displays OSPFv3 SPF calculation statistics.

	Command or Action	Purpose
Step 12	**show ospfv3** [*process-id*] [*address-family*] **summary-prefix** **Example:** `Device# show ospfv3 summary-prefix`	Displays a list of all summary address redistribution information configured under an OSPFv3 process.
Step 13	**show ospfv3** [*process-id*] [*address-family*] **timers rate-limit** **Example:** `Device# show ospfv3 timers rate-limit`	Displays all of the LSAs in the rate limit queue.
Step 14	**show ospfv3** [*process-id*] [*address-family*] **traffic**[*interface-type interface-number*] **Example:** `Device# show ospfv3 traffic`	Displays OSPFv3 traffic statistics.
Step 15	**show ospfv3** [*process-id*] [*address-family*] **virtual-links** **Example:** `Device# show ospfv3 virtual-links`	Displays parameters and the current state of OSPFv3 virtual links.

Configuration Examples for Load Balancing in OSPFv3

Example: Configuring the OSPFv3 Device Process

```
Device# show ospfv3 database
            OSPFv3 Device with ID (172.16.4.4) (Process ID 1)
                Device Link States (Area 0)
ADV Device         Age         Seq#        Fragment ID   Link count  Bits
172.16.4.4         239         0x80000003  0             1           B
172.16.6.6         239         0x80000003  0             1           B
            Inter Area Prefix Link States (Area 0)
ADV Device         Age         Seq#        Prefix
172.16.4.4         249         0x80000001  FEC0:3344::/32
172.16.4.4         219         0x80000001  FEC0:3366::/32
172.16.6.6         247         0x80000001  FEC0:3366::/32
172.16.6.6         193         0x80000001  FEC0:3344::/32
172.16.6.6         82          0x80000001  FEC0::/32
            Inter Area Device Link States (Area 0)
ADV Device         Age         Seq#        Link ID    Dest DevID
172.16.4.4         219         0x80000001  50529027   172.16.3.3
172.16.6.6         193         0x80000001  50529027   172.16.3.3

            Link (Type-8) Link States (Area 0)
ADV Device         Age         Seq#        Link ID    Interface
172.16.4.4         242         0x80000002  14         PO4/0
172.16.6.6         252         0x80000002  14         PO4/0
```

```
            Intra Area Prefix Link States (Area 0)
ADV Device         Age         Seq#        Link ID    Ref-lstype  Ref-LSID
172.16.4.4         242         0x80000002  0          0x2001      0
172.16.6.6         252         0x80000002  0          0x2001      0
```

```
Device# show ospfv3 neighbor

OSPFv3 Device with ID (10.1.1.1) (Process ID 42)
Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
10.4.4.4        1     FULL/  -        00:00:39    12              vm1
OSPFv3 Device with ID (10.2.1.1) (Process ID 100)
Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
10.5.4.4          1     FULL/  -         00:00:35    12               vm1
```

Example: Forcing SPF Configuration

The following example shows how to trigger SPF to redo the SPF and repopulate the routing tables:

```
clear ipv6 ospf force-spf
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
IPv6 Routing: OSPFv3	"*Configuring OSPF*" module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: OSPFv3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 204: Feature Information for IPv6 Routing: OSPFv3

Feature Name	Releases	Feature Information
IPv6 Routing: OSPFv3	Cisco IOS Release 15.2(6)E	OSPF version 3 for IPv6 expands on OSPF version 2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.

Table 205: Feature Information for IPv6 Routing: OSPFv3

Feature Name	Releases	Feature Information
IPv6 Routing: OSPFv3	Cisco IOS Release 17.4	This feature was introduced.

CHAPTER 195

IPv6 Routing: OSPFv3 Authentication Support with IPsec

In order to ensure that Open Shortest Path First version 3 (OSPFv3) packets are not altered and re-sent to the device, OSPFv3 packets must be authenticated. OSPFv3 uses the IPsec secure socket API to add authentication to OSPFv3 packets. This API supports IPv6.

Prerequisites for IPv6 Routing: OSPFv3 Authentication Support with IPsec

Configure the IP Security (IPsec) secure socket application program interface (API) on OSPFv3 in order to enable authentication and encryption.

Restrictions for IPv6 Routing: OSPFv3 Authentication Support with IPsec

The OSPF for IPv6(OSPFv3) Authentication Support with IPsec feature is not supported on the IP BASE license package. The Advanced Enterprise Services package license must be used.

Information About IPv6 Routing: OSPFv3 Authentication Support with IPsec

OSPFv3 Authentication Support with IPsec

In order to ensure that OSPFv3 packets are not altered and re-sent to the device, causing the device to behave in a way not desired by its system administrators, OSPFv3 packets must be authenticated. OSPFv3 uses the IPsec secure socket API to add authentication to OSPFv3 packets. This API supports IPv6.

OSPFv3 requires the use of IPsec to enable authentication. Crypto images are required to use authentication, because only crypto images include the IPsec API needed for use with OSPFv3.

In OSPFv3, authentication fields have been removed from OSPFv3 packet headers. When OSPFv3 runs on IPv6, OSPFv3 requires the IPv6 authentication header (AH) or IPv6 ESP header to ensure integrity, authentication, and confidentiality of routing exchanges. IPv6 AH and ESP extension headers can be used to provide authentication and confidentiality to OSPFv3.

To use the IPsec AH, you must enable the **ipv6 ospf authentication** command. To use the IPsec ESP header, you must enable the **ipv6 ospf encryption** command. The ESP header may be applied alone or in combination with the AH, and when ESP is used, both encryption and authentication are provided. Security services can be provided between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host.

To configure IPsec, you configure a security policy, which is a combination of the security policy index (SPI) and the key (the key is used to create and validate the hash value). IPsec for OSPFv3 can be configured on an interface or on an OSPFv3 area. For higher security, you should configure a different policy on each interface configured with IPsec. If you configure IPsec for an OSPFv3 area, the policy is applied to all of the interfaces in that area, except for the interfaces that have IPsec configured directly. Once IPsec is configured for OSPFv3, IPsec is invisible to you.

The secure socket API is used by applications to secure traffic. The API needs to allow the application to open, listen, and close secure sockets. The binding between the application and the secure socket layer also allows the secure socket layer to inform the application of changes to the socket, such as connection open and close events. The secure socket API is able to identify the socket; that is, it can identify the local and remote addresses, masks, ports, and protocol that carry the traffic requiring security.

Each interface has a secure socket state, which can be one of the following:

- NULL: Do not create a secure socket for the interface if authentication is configured for the area.
- DOWN: IPsec has been configured for the interface (or the area that contains the interface), but OSPFv3 either has not requested IPsec to create a secure socket for this interface, or there is an error condition.
- GOING UP: OSPFv3 has requested a secure socket from IPsec and is waiting for a CRYPTO_SS_SOCKET_UP message from IPsec.
- UP: OSPFv3 has received a CRYPTO_SS_SOCKET_UP message from IPsec.
- CLOSING: The secure socket for the interface has been closed. A new socket may be opened for the interface, in which case the current secure socket makes the transition to the DOWN state. Otherwise, the interface will become UNCONFIGURED.
- UNCONFIGURED: Authentication is not configured on the interface.

OSPFv3 will not send or accept packets while in the DOWN state.

How to Configure IPv6 Routing: OSPFv3 Authentication Support with IPsec

Configuring IPsec on OSPFv3

Once you have configured OSPFv3 and decided on your authentication, you must define the security policy on each of the devices within the group. The security policy consists of the combination of the key and the SPI. To define a security policy, you must define an SPI and a key.

You can configure an authentication or encryption policy either on an interface or for an OSPFv3 area. When you configure for an area, the security policy is applied to all of the interfaces in the area. For higher security, use a different policy on each interface.

You can configure authentication and encryption on virtual links.

Defining Authentication on an Interface

Before you begin

Before you configure IPsec on an interface, you must configure OSPFv3 on that interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ospfv3 authentication** {**ipsec** *spi*} {**md5** | **sha1**} { *key-encryption-type key*} | **null**
   - **ipv6 ospf authentication** {**null** | **ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] [*key*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface type and number, and places the device in interface configuration mode. **Note** You should configure the OSPFv3 authentication of the VLAN interface, instead of the physical interface. See the below example: `Device(config)# interface VLAN 60`

Step 4	Do one of the following:
	• **ospfv3 authentication** {**ipsec** *spi*} {**md5** | **sha1**} { *key-encryption-type key*} | **null**
	• **ipv6 ospf authentication** {**null** | **ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] [*key*]}
	Example:
	`Device(config-if)# ospfv3 authentication md5 0 275761340947681324733020312097 27`
	Example:
	Or
	`Device(config-if)# ipv6 ospf authentication ipsec spi 500 md5 1234567890abcdef1234567890abcdef`
		Specifies the authentication type for an interface.

Defining Authentication in an OSPFv3 Area

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **area** *area-id* **authentication ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] *key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 router ospf** *process-id* **Example:** `Device(config)# ipv6 router ospf 1`	Enables OSPFv3 router configuration mode.
Step 4	**area** *area-id* **authentication ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] *key* **Example:** `Device(config-rtr)# area 1 authentication ipsec spi 678 md5 1234567890ABCDEF1234567890ABCDEF`	Enables authentication in an OSPFv3 area.

Configuration Examples for IPv6 Routing: OSPFv3 Authentication Support with IPsec

Example: Defining Authentication on an Interface

The following example shows how to define authentication on Ethernet interface 0/0:

```
interface Ethernet0/0
 ipv6 enable
 ipv6 ospf 1 area 0
 ipv6 ospf authentication ipsec spi 500 md5 1234567890ABCDEF1234567890ABCDEF
interface Ethernet0/0
 ipv6 enable
 ipv6 ospf authentication null
 ipv6 ospf 1 area 0
```

Example: Defining Authentication in an OSPFv3 Area

The following example shows how to define authentication on OSPFv3 area 0:

```
ipv6 router ospf 1
 router-id 10.11.11.1
 area 0 authentication ipsec spi 1000 md5 1234567890ABCDEF1234567890ABCDEF
```

Additional References for IPv6 Routing: OSPFv3 Authentication Support with IPsec

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
IPv6 Routing: OSPF for IPv6 Authentication Support with IPsec	"*Configuring OSPF* " module in *IP Routing: OSPF Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: OSPFv3 Authentication Support with IPsec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 206: Feature Information for IPv6 Routing: OSPF for IPv6 Authentication Support with IPsec

Feature Name	Releases	Feature Information
IPv6 Routing: OSPF for IPv6 Authentication Support with IPsec	XE 3.14S	OSPFv3 uses the IPsec secure socket API to add authentication to OSPFv3 packets. The following commands were introduced or modified: **area authentication (IPv6)**, **ipv6 ospf authentication**, **ipv6 router ospf**, **ospfv3 authentication**.

Table 207: Feature Information for IPv6 Routing: OSPF for IPv6 Authentication Support with IPsec

Feature Name	Releases	Feature Information
IPv6 Routing: OSPF for IPv6 Authentication Support with IPsec	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 196

OSPFv2 Cryptographic Authentication

To prevent unauthorized or invalid routing updates in your network, Open Shortest Path First version 2 (OSPFv2) protocol packets must be authenticated.

There are two methods of authentication that are defined for OSPFv2: plain text authentication and cryptographic authentication. This module describes how to configure cryptographic authentication using the Hashed Message Authentication Code - Secure Hash Algorithm (HMAC-SHA). OSPFv2 specification (RFC 2328) allows only the Message-Digest 5 (MD5) algorithm for cryptographic authentication. However, RFC 5709 (OSPFv2 HMAC-SHA Cryptographic Authentication) allows OSPFv2 to use HMAC-SHA algorithms for cryptographic authentication.

Prerequisites for OSPFv2 Cryptographic Authentication

Ensure that Open Shortest Path First version 2 (OSPFv2) is configured on your network.

Information About OSPFv2 Cryptographic Authentication

Configuring OSPFv2 Cryptographic Authentication

The OSPFv2 Cryptographic Authentication feature allows you to configure a key chain on the OSPF interface to authenticate OSPFv2 packets by using HMAC-SHA algorithms. You can use an existing key chain that is being used by another protocol, or you can create a key chain specifically for OSPFv2.

A key chain is a list of keys. Each key consists of a key string, which is also called the password or passcode. A key-string is essential for a key to be operational. Each key is identified by a unique key ID. To authenticate the OSPFv2 packets, it is essential that the cryptographic authentication algorithm be configured with a key. OSPFv2 supports keys with key IDs ranging from 1 to 255. The combination of the cryptographic authentication algorithm and the key is known as a Security Association (SA).

The authentication key on a key chain is valid for a specific time period called lifetime. An SA has the following configurable lifetimes:

- Accept lifetime
- Send lifetime

While adding a new key, the Send lifetime is set to a time in the future so that the same key can be configured on all devices in the network before the new key becomes operational. Old keys are removed only after the new key is operational on all devices in the network. When packets are received, the key ID is used to fetch the data for that key. The packet is verified using the cryptographic authentication algorithm and the configured key ID. If the key ID is not found, the packet is dropped.

Note When key chain has more than one key, OSPF selects the key that has the maximum life time. Key having an infinite lifetime is preferred. If keys have the same lifetime, then key with the higher key ID is preferred.

Use the **ip ospf authentication key-chain** command to configure key chains for OSPFv2 cryptographic authentication.

Note If OSPFv2 is configured to use a key chain, all MD5 keys that were previously configured using the **ip ospf message-digest-key** command are ignored.

Note If you receive packets that come in a non-decreasing sequence, the system displays an authentication error. This is not an error, and you can ignore this authentication error message. No other action is required from your end.

How to Configure OSPFv2 Cryptographic Authentication

Defining a Key Chain

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name*
4. **key** *key-id*
5. **key-string** *name*
6. **cryptographic-algorithm** *name*
7. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name* **Example:** `Device(config)# key chain sample1`	Specifies the key chain name and enters key-chain configuration mode.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Specifies the key identifier and enters key-chain key configuration mode. The range is from 1 to 255.
Step 5	**key-string** *name* **Example:** `Device(config-keychain-key)# key-string string1`	Specifies the key string.
Step 6	**cryptographic-algorithm** *name* **Example:** `Device(config-keychain-key)# cryptographic-algorithm hmac-sha-256`	Configures the key with the specified cryptographic algorithm.
Step 7	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime local 10:00:00 5 July 2013 infinite`	Sets the time period during which an authentication key on a key chain is valid to be sent during key exchange with another device.
Step 8	**end** **Example:** `Device(config-keychain-key)# end`	Exits key-chain key configuration mode and returns to privileged EXEC mode.

Defining Authentication on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf authentication key-chain** *name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet0/0/0`	Specifies an interface type and number and enters interface configuration mode.
Step 4	**ip ospf authentication key-chain** *name* **Example:** `Device(config-if)# ip ospf authentication key-chain ospf1`	Specifies the key chain for an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for OSPFv2 Cryptographic Authentication

Example: Defining a Key Chain

The following example shows how to configure a key chain:

```
Device> enable
Device# configure terminal
Device(config)# key chain sample1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string ThisIsASampleKey12345
Device(config-keychain-key)# cryptographic-algorithm hmac-sha-256
Device(config-keychain-key)# send-lifetime local 10:00:00 5 July 2013 infinite
Device(config-keychain-key)# end
```

Example: Verifying a Key Chain

The following sample output from the **show key chain** command displays the key chain information:

```
Device# show key chain Key-chain sample1

     key 1 -- text "ThisIsASampleKey12345"
         accept lifetime (always valid) - (always valid) [valid now]
         send lifetime (10:00:00 PDT Jul 5 2013) - (infinite)
```

The table below describes the significant fields in the output:

Table 208: show ip ospf interface Field Descriptions

Field	Description
key	Status of the configured key.
accept lifetime	The time interval within which the device accepts the key during key exchange with another device.
send lifetime	The time interval within which the device sends the key during a key exchange with another device.

Example: Defining Authentication on an Interface

The following example shows how to define authentication on Gigabit Ethernet interface 0/0/0:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet0/0/0
Device (config-if)# ip ospf authentication key-chain sample1
Device (config-if)# end
```

Example: Verifying Authentication on an Interface

The following sample output of the **show ip ospf interface** command displays the cryptographic key information:

```
Device# show ip ospf interface GigabitEthernet0/0/0

GigabitEthernet0/0/0 is up, line protocol is up
  Internet Address 192.168.8.2/24, Area 1, Attached via Interface Enable
  Process ID 1, Router ID 10.1.1.8, Network Type BROADCAST, Cost: 10
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           10        no          no            Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.1.8, Interface address 192.168.8.2
  Backup Designated router (ID) 10.1.1.9, Interface address 192.168.8.9
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Can be protected by per-prefix Loop-Free FastReroute
  Can be used for per-prefix Loop-Free FastReroute repair paths
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.1.9  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
  Cryptographic authentication enabled
    Sending SA: Key 25, Algorithm HMAC-SHA-256 - key chain sample1
```

The table below describes the significant fields in the output:

Table 209: show ip ospf interface Field Descriptions

Field	Description
GigabitEthernet	Status of the physical link and operational status of the protocol.
Internet Address	Interface IP address, subnet mask, and area address.
Area	OSPF area.
Process ID	OSPF process ID.
Cost	Administrative cost assigned to the interface.
Topology-MTID	MTR topology Multitopology Identifier (MTID) is a number assigned so that the protocol can identify the topology associated with information that it sends to its peers.
Transmit Delay	Transmit delay (in seconds), interface state, and router priority.
State	Operational state of the interface.

Field	Description
Designated Router	Designated router ID and respective interface IP address.
Backup Designated router	Backup designated router ID and respective interface IP address.
Timer intervals configured	Configuration of timer intervals.
Neighbor Count	Count of network neighbors and list of adjacent neighbors.
Cryptographic authentication	Status of cryptographic authentication.
Sending SA	Status of the sending SA (Security Association). Key, cryptographic algorithm, and key chain used.

Additional References for OSPFv2 Cryptographic Authentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference

Standards and RFCs

Standard	Title
RFC 2328	OSPF Version 2 , April 1998
RFC 5709	OSPFv2 HMAC-SHA Cryptographic Authentication , October 2009

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv2 Cryptographic Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 210: Feature Information for OSPFv2 Cryptographic Authentication

Feature Name	Releases	Feature Information
OSPFv2 Cryptographic Authentication	15.4(1)T	The OSPFv2 Cryptographic Authentication feature prevents unauthorized or invalid routing updates in your network by authenticating Open Shortest Path First version 2 (OSPFv2) protocol packets using HMAC-SHA algorithms. The following command was modified: **ip ospf authentication**.

Table 211: Feature Information for OSPFv2 Cryptographic Authentication

Feature Name	Releases	Feature Information
OSPFv2 Cryptographic Authentication	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 197

OSPFv3 External Path Preference Option

The Open Shortest Path First version 3 (OSPFv3) external path preference option feature provides a way to calculate external path preferences per RFC 5340.

Information About OSPFv3 External Path Preference Option

OSPFv3 External Path Preference Option

Per RFC 5340, the following rules indicate which paths are preferred when multiple intra-AS paths are available to ASBRs or forwarding addresses:

- Intra-area paths using nonbackbone areas are always the most preferred.
- The other paths, intraarea backbone paths and interarea paths, are of equal preference.

These rules apply when the same ASBR is reachable through multiple areas, or when trying to decide which of several AS-external-LSAs should be preferred. In the former case the paths all terminate at the same ASBR, and in the latter the paths terminate at separate ASBRs or forwarding addresses. In either case, each path is represented by a separate routing table entry. This feature applies only when RFC 1583 compatibility is set to disabled using the **no compatibility rfc1583** command (RFC 5340 provides an update to RFC 1583).

Caution To minimize the chance of routing loops, set identical RFC compatibility for all OSPF routers in an OSPF routing domain.

How to Calculate OSPFv3 External Path Preference Option

Calculating OSPFv3 External Path Preferences per RFC 5340

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **no compatible rfc1583**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**no compatible rfc1583** **Example:** `Device(config-router)# no compatible rfc1583`	Changes the method used to calculate external path preferences per RFC 5340.

Configuration Examples for OSPFv3 External Path Preference Option

Example: Calculating OSPFv3 External Path Preferences per RFC 5340

```
show ospfv3

Routing Process "ospfv3 1" with ID 10.1.1.1
```

```
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Number of external LSA 0. Checksum Sum 0x000000
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Reference bandwidth unit is 100 mbps
RFC 1583 compatibility disabled
   Area BACKBONE(0) (Inactive)
       Number of interfaces in this area is 1
       SPF algorithm executed 1 times
       Number of LSA 1. Checksum Sum 0x00D03D
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
OSPFv3 External Path Preference Option	"*Configuring OSPF* " module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 External Path Preference Option

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 212: Feature Information for OSPFv3 External Path Preference Option

Feature Name	Releases	Feature Information
OSPFv3 External Path Preference Option	Cisco IOS XE Release 3.4S	This feature provides a way to calculate external path preferences per RFC 5340. The following commands were introduced or modified: **compatible rfc1583**, **show ospfv3**.

Table 213: Feature Information for OSPFv3 External Path Preference Option

Feature Name	Releases	Feature Information
OSPFv3 External Path Preference Option	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 198

OSPFv3 Graceful Restart

The graceful restart feature in Open Shortest Path First version 3 (OSPFv3) allows nonstop data forwarding along routes that are already known while the OSPFv3 routing protocol information is being restored.

Information About OSPFv3 Graceful Restart

OSPFv3 Graceful Restart

The graceful restart feature in OSPFv3 allows nonstop data forwarding along routes that are already known while the OSPFv3 routing protocol information is being restored. A device can participate in graceful restart either in restart mode (such as in a graceful-restart-capable device) or in helper mode (such as in a graceful-restart-aware device).

To perform the graceful restart function, a device must be in high availability (HA) stateful switchover (SSO) mode (that is, dual Route Processor (RP)). A device capable of graceful restart will perform the graceful restart function when the following failures occur:

- A RP failure that results in switchover to standby RP
- A planned RP switchover to standby RP

The graceful restart feature requires that neighboring devices be graceful-restart aware.

For further information about SSO and nonstop forwarding (NSF), see the Stateful Switchover and Cisco Nonstop Forwarding documents.

How to Enable OSPFv3 Graceful Restart

Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Capable Router

The task can be performed in Cisco IOS XE 3.4S and later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **graceful-restart** [**restart-interval** *interval*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Router(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**graceful-restart** [**restart-interval** *interval*] **Example:** `Router(config-rtr)# graceful-restart`	Enables the OSPFv3 graceful restart feature on a graceful-restart-capable router.

Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Capable Router

The task can be performed in releases prior to Cisco IOS XE Release 3.4S.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **graceful-restart** [**restart-interval** *interval*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router ospf** *process-id* **Example:** `Router(config)# ipv6 router ospf 1`	Enables OSPFv3 router configuration mode.
Step 4	**graceful-restart** [**restart-interval** *interval*] **Example:** `Router(config-rtr)# graceful-restart`	Enables the OSPFv3 graceful restart feature on a graceful-restart-capable router.

Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Aware Router

The task can be performed in Cisco IOS XE Release 3.4S and later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **graceful-restart helper** {**disable** | **strict-lsa-checking**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router ospfv3** [*process-id*] **Example:** `Router(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**graceful-restart helper** {**disable** | **strict-lsa-checking** **Example:** `Router(config-rtr)# graceful-restart helper strict-lsa-checking`	Enables the OSPFv3 graceful restart feature on a graceful-restart-aware router.

Example:

What to do next

Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Aware Router

The task can be performed in releases prior to Cisco IOS XE Release 3.4S.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **graceful-restart helper** {**disable** | **strict-lsa-checking**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router ospf** *process-id* **Example:** `Router(config)# ipv6 router ospf 1`	Enables OSPFv3 router configuration mode.

	Command or Action	Purpose
Step 4	**graceful-restart helper** {**disable** | **strict-lsa-checking** **Example:** `Router(config-rtr)# graceful-restart helper strict-lsa-checking`	Enables the OSPFv3 graceful restart feature on a graceful-restart-aware router.

Example:

What to do next

Configuration Examples for OSPFv3 Graceful Restart

Example: Enabling OSPFv3 Graceful Restart

```
Router# show ipv6 ospf graceful-restart
Routing Process "ospf 1"
 Graceful Restart enabled
    restart-interval limit: 120 sec, last restart 00:00:15 ago (took 36 secs)
  Graceful Restart helper support enabled
  Router status : Active
  Router is running in SSO mode
  OSPF restart state : NO_RESTART
  Router ID 10.1.1.1, checkpoint Router ID 10.0.0.0
```

The following example shows OSPFv3 information with graceful-restart helper support enabled on a graceful-restart-aware router.

```
Router# show ospfv3
 Routing Process "ospfv3 1" with ID 10.0.0.1
 Supports IPv6 Address Family
 Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 Graceful restart helper support enabled
 Reference bandwidth unit is 100 mbps
 Relay willingness value is 128
 Pushback timer value is 2000 msecs
 Relay acknowledgement timer value is 1000 msecs
 LSA cache Disabled : current count 0, maximum 1000
 ACK cache Disabled : current count 0, maximum 1000
```

```
Selective Peering is not enabled
Hello requests and responses will be sent multicast
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Stateful switchover and Cisco nonstop forwarding	*High Availability Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
OSPFv3 Graceful Restart	"*OSPF RFC 3623 Graceful Restart Helper Mode*" module
OSPFv3 Graceful Restart	"*Configuring OSPF* " module
OSPFv3 Graceful Restart	"*NSF-OSPF RFC 3623 OSPF Graceful Restart* " module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 Graceful Restart

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 214: Feature Information for OSPFv3 Graceful Restart

Feature Name	Releases	Feature Information
OSPFv3 Graceful Restart	Cisco IOS XE Release 2.1	The graceful restart feature in OSPFv3 allows nonstop data forwarding along routes that are already known while the OSPFv3 routing protocol information is being restored. The following commands were introduced or modified: **graceful-restart**, **graceful-restart helper**, **ipv6 router ospf**, **router ospfv3**, **show ipv6 ospf graceful-restart**, **show ospfv3 graceful-restart**.

Table 215: Feature Information for OSPFv3 Graceful Restart

Feature Name	Releases	Feature Information
OSPFv3 Graceful Restart	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 199

Graceful Shutdown Support for OSPFv3

This feature provides the ability to temporarily shut down an Open Shortest Path First version 3 (OSPFv3) process or interface in the least disruptive manner, and to notify its neighbors that it is going away. A graceful shutdown of a protocol can be initiated on all OSPFv3 interfaces or on a specific interface.

Information About Graceful Shutdown Support for OSPFv3

OSPFv3 Graceful Shutdown

The Graceful Shutdown for OSPFv3 feature provides the ability to temporarily shut down the OSPFv3 protocol in the least disruptive manner and to notify its neighbors that it is going away. All traffic that has another path through the network will be directed to that alternate path. A graceful shutdown of the OSPFv3 protocol can be initiated using the **shutdown** command in router configuration mode or in address family configuration mode.

This feature also provides the ability to shut down OSPFv3 on a specific interface. In this case, OSPFv3 will not advertise the interface or form adjacencies over it; however, all of the OSPFv3 interface configuration will be retained. To initiate a graceful shutdown of an interface, use the **ipv6 ospf shutdown** or the **ospfv3 shutdown** command in interface configuration mode.

How to Configure Graceful Shutdown Support for OSPFv3

Configuring Graceful Shutdown of the OSPFv3 Process

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. Do one of the following:
   - **ipv6 router ospf** *process-id*
   - **router ospfv3** *process-id*
4. **shutdown**
5. **end**
6. Do one of the following:
   - **show ipv6 ospf** [*process-id*]
   - **show ospfv3** [*process-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ipv6 router ospf** *process-id*
	• **router ospfv3** *process-id*
	Example:
	`Device(config)# ipv6 router ospf 1`
	Example:
	`Device(config)# router ospfv3 101`
		Enables OSPFv3 routing and enters router configuration mode.

Step 4	**shutdown** **Example:** `Device(config-router)# shutdown`	Shuts down the selected interface.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 6	Do one of the following: • **show ipv6 ospf** [*process-id*] • **show ospfv3** [*process-id*] **Example:** `Device# show ipv6 ospf` **Example:**	(Optional) Displays general information about OSPFv3 routing processes.

	Command or Action	Purpose
	`Device# show ospfv3`	

Configuring Graceful Shutdown of the OSPFv3 Process in Address-Family Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **address-family ipv6 unicast** [**vrf** *vrf-name*]
5. **shutdown**
6. **end**
7. **show ospfv3** [*process-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables router configuration mode for the IPv6 address family.
Step 4	**address-family ipv6 unicast** [**vrf** *vrf-name*] **Example:** `Device(config-router)#address-family ipv6`	Enters IPv6 address family configuration mode for OSPFv3.
Step 5	**shutdown** **Example:** `Device(config-router-af)# shutdown`	Shuts down the selected interface.
Step 6	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-router-af)# end`	
Step 7	**show ospfv3** [*process-id*] **Example:** `Device# show ospfv3`	(Optional) Displays general information about OSPFv3 routing processes.

Configuring OSPFv3 Graceful Shutdown of the OSPFv3 Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ipv6 ospf shutdown**
   - **ospfv3 shutdown**
5. **end**
6. **show ospfv3** *process-id* [*area-id*] [*address-family*] [**vrf** {*vrf-name* | ***** }] **interface** [*type number*] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet`	Configures an interface type and number and enters interface configuration mode.
Step 4	Do one of the following: • **ipv6 ospf shutdown** • **ospfv3 shutdown** **Example:**	Initiates an OSPFv3 protocol graceful shutdown at the interface level. • When the **ipv6 ospf shutdown** interface command is entered, the interface on which it is configured sends a link-state update advising its neighbors that is going

	Command or Action	Purpose

	`Device(config-if)# ipv6 ospf shutdown`
	Example:
	`Device(config-if)# ospfv3 process-id ipv6 shutdown`
		down, which allows those neighbors to begin routing OSPFv3 traffic around this device.

Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ospfv3** *process-id* [*area-id*] [*address-family*] [**vrf** {*vrf-name* | ***** }] **interface** [*type number*] [**brief**] **Example:** `Device# show ospfv3 1 interface`	(Optional) Displays OSPFv3-related interface information.

Configuration Examples for Graceful Shutdown Support for OSPFv3

Example: Configuring Graceful Shutdown of the OSPFv3 Process

The following example shows how to configure graceful shutdown of the OSPFv3 process in IPv6 router OSPF configuration mode configuration mode:

```
ipv6 router ospf 6
 router-id 10.10.10.10
 shutdown
```

The following example shows how to configure graceful shutdown of the OSPFv3 process in router OSPFv3 configuration mode:

```
!
router ospfv3 1
 shutdown
 !
 address-family ipv6 unicast
  exit-address-family
```

The following example shows how to configure graceful shutdown of the OSPFv3 process in address-family configuration mode:

```
!
router ospfv3 1
 !
 address-family ipv6 unicast
  shutdown
  exit-address-family
```

Example: Configuring Graceful Shutdown of the OSPFv3 Interface

The following example shows how to configure graceful shutdown of the OSPFv3 interface using the **ipv6 ospf shutdown** command:

```
!
interface Serial2/1
 no ip address
 ipv6 enable
 ipv6 ospf 6 area 0
 ipv6 ospf shutdown
 serial restart-delay 0
 end
```

The following example shows how to configure graceful shutdown of the OSPFv3 interface using the **ospfv3 shutdown** command:

```
!
interface Serial2/0
 ip address 10.10.10.10 255.255.255.0
 ip ospf 1 area 0
 ipv6 enable
 ospfv3 shutdown
 ospfv3 1 ipv6 area 0
 serial restart-delay 0
 end
```

Additional References for Graceful Shutdown Support for OSPFv3

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Graceful Shutdown Support for OSPFv3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 216: Feature Information for Graceful Shutdown Support for OSPFv3

Feature Name	Releases	Feature Information
Graceful Shutdown Support for OSPFv3	Cisco IOS XE Release 3.8	This feature provides the ability to temporarily shut down an Open Shortest Path First version 3 (OSPFv3) process or interface in the least disruptive manner, and to notify its neighbors that it is going away. A graceful shutdown of a protocol can be initiated on all OSPFv3 interfaces or on a specific interface. The following commands were introduced: • **ipv6 ospf shutdown** • **ospfv3 shutdown** • **shutdown (router ospfv3)**

Table 217: Feature Information for Graceful Shutdown Support for OSPFv3

Feature Name	Releases	Feature Information
Graceful Shutdown Support for OSPFv3	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 200

OSPF Stub Router Advertisement

The OSPF Stub Router Advertisement feature allows you to bring a new router into a network without immediately routing traffic through the new router and allows you to gracefully shut down or reload a router without dropping packets that are destined for other networks.

Information About OSPF Stub Router Advertisement

OSPF Stub Router Advertisement Functionality

The OSPF Stub Router Advertisement feature allows you to bring a new router into a network without immediately routing traffic through the new router and allows you to gracefully shut down or reload a router without dropping packets that are destined for other networks. This feature introduces three configuration options that allow you to configure a router that is running the Open Shortest Path First (OSPF) protocol to advertise a maximum or infinite metric to all neighbors.

When any of these three configuration options are enabled on a router, the router will originate link-state advertisements (LSAs) with a maximum metric (LSInfinity: 0xFFFF) through all nonstub links. The advertisement of a maximum metric causes other routers to assign a cost to the new router that is higher than the cost of using an alternate path. Because of the high cost assigned to paths that pass through the new router, other routers will not use a path through the new router as a transit path to forward traffic that is destined for other networks, which allows switching and routing functions to be up and running and routing tables to converge before transit traffic is routed through this router.

Note Directly connected links in a stub network are not affected by the configuration of a maximum or infinite metric because the cost of a stub link is always set to the output interface cost.

Maximum Metric Allows Routing Tables to Converge

Two configuration options introduced by the OSPF Stub Router Advertisement feature allow you to bring a new router into a network without immediately routing traffic through the new router. These configuration options are useful because Interior Gateway Protocols (IGPs) converge very quickly upon a router during startup or after a reload, often before Border Gateway Protocol (BGP) routing tables have completely converged. If neighbor routers forward traffic through a router while that router is building BGP routing tables, packets that have been received for other destinations may be dropped. Advertising a maximum metric during startup will allow routing tables to converge before traffic that is destined for other networks is sent through the router.

The following two configuration options enable a router to advertise a maximum metric at startup:

- You can configure a timer to advertise a maximum metric when the router is started or reloaded. When this option is configured, the router will advertise a maximum metric, which forces neighbor routers to select alternate paths until the timer expires. When the timer expires, the router will advertise accurate (normal) metrics, and other routers will send traffic to this router depending on the cost. The configurable range of the timer is from 5 to 86,400 seconds.
- You can configure a router to advertise a maximum metric at startup until BGP routing tables converge or until the default timer expires (600 seconds). Once BGP routing tables converge or the default timer expires, the router will advertise accurate (normal) metrics and other routers will send traffic to this router, depending on the cost.

Maximum Metric Allows Graceful Shutdown of a Router

The third configuration option introduced by the OSPF Stub Router Advertisement feature allows you to gracefully remove a router from the network by advertising a maximum metric through all links, which allows other routers to select alternate paths for transit traffic to follow before the router is shut down. There are many situations where you may need to remove a router from the network. If a router is removed from a network and neighbor routers cannot detect that the physical interface is down, neighbors will need to wait for dead timers to expire before the neighbors will remove the adjacency and routing tables will reconverge. This situation may occur when there is a switch between other routers and the router that is shut down. Packets may be dropped while the neighbor routing tables reconverge.

When this third option is configured, the router advertises a maximum metric, which allows neighbor routers to select alternate paths before the router is shut down. This configuration option could also be used to remove a router that is in a critical condition from the network without affecting traffic that is destined for other networks.

Note You should not save the running configuration of a router when it is configured for a graceful shutdown because the router will continue to advertise a maximum metric after it is reloaded.

Benefits of OSPF Stub Router Advertisement

Improved Stability and Availability

Advertising a maximum metric through all links at startup or during a reload will prevent neighbor routers from using a path through the router as a transit path, thereby reducing the number of packets that are dropped and improving the stability and availability of the network.

Graceful Removal from the Network

Advertising a maximum metric before shutdown allows other routers to select alternate paths before the transit path through a router becomes inaccessible.

How to Configure OSPF Stub Router Advertisement

The following tasks configure OSPF to advertise a maximum metric. This feature has three different configuration options. All tasks are optional and should be individually configured.

Configuring Advertisement on Startup

SUMMARY STEPS

1. Router(config)# **router ospf** *process-id*
2. Router(config-router)# **max-metric router-lsa on-startup** *announce-time*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router ospf** *process-id*	Places the router in router configuration mode and enables an OSPF routing process.
Step 2	Router(config-router)# **max-metric router-lsa on-startup** *announce-time*	Configures OSPF to advertise a maximum metric during startup for a configured period of time. The *announce-time* argument is a configurable timer that must follow the **on-startup** keyword to be configured. There is no default timer value. The configurable time range is from 5 to 86,400 seconds.

Configuring Advertisement Until Routing Tables Converge

SUMMARY STEPS

1. Router(config)# **router ospf** *process-id*
2. Router(config-router)# **max-metric router-lsa on-startup wait-for-bgp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router ospf** *process-id*	Places the router in router configuration mode and enables an OSPF routing process.
Step 2	Router(config-router)# **max-metric router-lsa on-startup wait-for-bgp**	Configures OSPF to advertise a maximum metric until BGP routing tables have converged or until the default timer has expired. The **wait-for-bgp** keyword must follow the **on-startup** keyword to be configured. The default timer value is 600 seconds.

Configuring Advertisement for a Graceful Shutdown

SUMMARY STEPS

1. Router(config)# **router ospf**process-id
2. Router(config-router)# **max-metric router-lsa**
3. Router(config-router)# **end**
4. Router# **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router ospf**process-id	Places the router in router configuration mode and enables an OSPF routing process.
Step 2	Router(config-router)# **max-metric router-lsa**	Configures OSPF to advertise a maximum metric until the router is shut down.
Step 3	Router(config-router)# **end**	Ends configuration mode and places the router in privileged EXEC mode.
Step 4	Router# **show ip ospf**	Displays general information about OSPF routing processes. • Use the **show ip ospf** command to verify that the **max-metric router-lsa** command has been enabled before the router is shut down or reloaded.

What to do next

Note Do not save the running configuration of a router when it is configured for a graceful shutdown because the router will continue to advertise a maximum metric after it is reloaded.

Verifying the Advertisement of a Maximum Metric

To verify that the advertisement of a maximum metric has been configured correctly, use the **show ip ospf**or **show ip ospf database**command.

The output of the **show ip ospf**command will display the condition, state, and remaining time delay of the advertisement of a maximum metric, depending on which options were configured with the **max-metric router-lsa** command.

The following sample output is similar to the output that will be displayed when the **on-startup**keyword and *announce-time* argument are configured with the **max-metric router-lsa** command:

```
Router# show ip ospf
Routing Process "ospf 1998" with ID 10.18.134.155
     Supports only single TOS(TOS0) routes
     Supports opaque LSA
     It is an area border and autonomous system boundary router
     Redistributing External Routes from,
        static, includes subnets in redistribution
     Originating router-LSAs with maximum metric, Time remaining: 00:01:18
        Condition: on startup for 300 seconds, State: active
     SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
     Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
     Number of external LSA 7. Checksum Sum 0x47261
     Number of opaque AS LSA 0. Checksum Sum 0x0
     Number of DCbitless external and opaque AS LSA 0
     Number of DoNotAge external and opaque AS LSA 0
     Number of areas in this router is 2. 1 normal 0 stub 1 nssa
     External flood list length 0
        Area BACKBONE(0)
           Number of interfaces in this area is 1
           Area has no authentication
           SPF algorithm executed 3 times
           Area ranges are
           Number of LSA 8. Checksum Sum 0x474AE
           Number of opaque link LSA 0. Checksum Sum 0x0
```

The following sample output is similar to the output that will be displayed when the **on-startup**and **wait-for-bgp** keywords are configured with the **max-metric router-lsa** command:

```
Router# show ip ospf
Routing Process "ospf 1998" with ID 10.18.134.155
     Supports only single TOS(TOS0) routes
     Supports opaque LSA
     It is an area border and autonomous system boundary router
     Redistributing External Routes from,
        static, includes subnets in redistribution
     Originating router-LSAs with maximum metric, Time remaining: 00:01:18
        Condition: on startup while BGP is converging, State: active
     SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
     Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
     Number of external LSA 7. Checksum Sum 0x47261
     Number of opaque AS LSA 0. Checksum Sum 0x0
     Number of DCbitless external and opaque AS LSA 0
     Number of DoNotAge external and opaque AS LSA 0
     Number of areas in this router is 2. 1 normal 0 stub 1 nssa
     External flood list length 0
        Area BACKBONE(0)
           Number of interfaces in this area is 1
           Area has no authentication
           SPF algorithm executed 3 times
```

```
        Area ranges are
        Number of LSA 8. Checksum Sum 0x474AE
        Number of opaque link LSA 0. Checksum Sum 0x0
```

The following sample output is similar to the output that will be displayed when the **max-metric router-lsa** command is configured without any keywords or arguments:

```
Router# show ip ospf
Routing Process "ospf 1998" with ID 10.18.134.155
     Supports only single TOS(TOS0) routes
     Supports opaque LSA
     It is an area border and autonomous system boundary router
     Redistributing External Routes from,
        static, includes subnets in redistribution
     Originating router-LSAs with maximum metric
        Condition: always, State: active
     SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
     Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
     Number of external LSA 7. Checksum Sum 0x47261
     Number of opaque AS LSA 0. Checksum Sum 0x0
     Number of DCbitless external and opaque AS LSA 0
     Number of DoNotAge external and opaque AS LSA 0
     Number of areas in this router is 2. 1 normal 0 stub 1 nssa
     External flood list length 0
        Area BACKBONE(0)
           Number of interfaces in this area is 1
           Area has no authentication
           SPF algorithm executed 3 times
           Area ranges are
           Number of LSA 8. Checksum Sum 0x474AE
           Number of opaque link LSA 0. Checksum Sum 0x0
```

The output of the **show ip ospf database**command will display information about OSPF LSAs and indicate if the router is announcing maximum cost links. The following sample output is similar to the output that will be displayed when any form of the **max-metric router-lsa** command is configured:

```
Router# show ip ospf database
      Exception Flag: Announcing maximum link costs
      LS age: 68
      Options: (No TOS-capability, DC)
      LS Type: Router Links
      Link State ID: 172.18.134.155
      Advertising Router: 172.18.134.155
      LS Seq Number: 80000002
      Checksum: 0x175D
      Length: 60
      Area Border Router
      AS Boundary Router
      Number of Links: 3

        Link connected to: a Transit Network
        (Link ID) Designated Router address: 192.168.1.11
        (Link Data) Router Interface address: 192.168.1.14
         Number of TOS metrics: 0
          TOS 0 Metrics: 65535 (metric used for local calculation: 10)

        Link connected to: a Transit Network
        (Link ID) Designated Router address: 10.1.145.11
        (Link Data) Router Interface address: 10.1.145.14
         Number of TOS metrics: 0
          TOS 0 Metrics: 65535 (metric used for local calculation: 10)
```

```
Link connected to: a Stub Network
(Link ID) Network/subnet number: 10.11.12.0
(Link Data) Network Mask: 255.255.255.0
 Number of TOS metrics: 0
  TOS 0 Metrics: 1
```

Monitoring and Maintaining OSPF Stub Router Advertisement

Command	Purpose
`Router# show ip ospf`	Displays general information about OSPF routing processes and provides information about the configuration settings and status of the OSPF Stub Router Advertisement feature.
`Router# show ip ospf database router`	Displays information about router LSAs, and indicates if a router is announcing maximum link costs.

Configuration Examples of OSPF Stub Router Advertisement

Example Advertisement on Startup

In the following example, a router that is running OSPF is configured to advertise a maximum metric at startup for 300 seconds:

```
Router(config)# router ospf 100
 Router(config-router)# max-metric router-lsa on-startup 300
```

Example Advertisement Until Routing Tables Converge

In the following example, a router that is running OSPF is configured to advertise a maximum metric until BGP routing tables converge or until the default timer expires (600 seconds):

```
Router(config)# router ospf 100
 Router(config-router)# max-metric router-lsa on-startup wait-for-bgp
```

Example Graceful Shutdown

In the following example, a router that is running OSPF is configured to advertise a maximum metric until the router is shut down:

```
Router(config)# router ospf 100
 Router(config-router)# max-metric router-lsa
 Router(config-router)# end
 Router# show ip ospf
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF	"Configuring OSPF" in the *IP Routing: OSPF Configuration Guide*.
OSPFv2 loop-free alternate fast reroute	"OSPFv2 Loop-Free Alternate Fast Reroute" in the *IP Routing: OSPF Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 5286	*Basic Specification for IP Fast Reroute: Loop-Free Alternates*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Stub Router Advertisement

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 218: Feature Information for OSPF Stub Router Advertisement

Feature Name	Releases	Feature Information
OSPF Stub Router Advertisement	Cisco IOS XE Release 2.1	The OSPF Stub Router Advertisement feature allows you to bring a new router into a network without immediately routing traffic through the new router and allows you to gracefully shut down or reload a router without dropping packets that are destined for other networks. The following commands are introduced or modified in the feature documented in this module: • **max-metric router-lsa** • **show ip ospf**

Table 219: Feature Information for OSPF Stub Router Advertisement

Feature Name	Releases	Feature Information
OSPF Stub Router Advertisement	Cisco IOS XE Release17.4	This feature was introduced.

CHAPTER 201

OSPF Update Packet-Pacing Configurable Timers

This module describes the OSPF Update Packet-Pacing Configurable Timers feature, which allows you to configure the rate at which OSPF LSA flood pacing, retransmission pacing, and group pacing updates occur.

Restrictions on OSPF Update Packet-Pacing Configurable Timers

Do not change the packet-pacing timers unless all other options to meet OSPF packet flooding requirements have been exhausted. Specifically, network operators should prefer summarization, stub area usage, queue tuning, and buffer tuning before changing the default timers. Furthermore, there are no guidelines for changing timer values; each OSPF deployment is unique and should be considered on a case-by-case basis. The network operator assumes risks that are associated with changing the default timer values.

Information About OSPF Update Packet-Pacing Configurable Timers

Functionality of the OSPF Update Packet-Pacing Timers

In rare situations, you might need to change Open Shortest Path First (OSPF) packet-pacing default timers to mitigate CPU or buffer utilization issues associated with flooding very large numbers of link-state advertisements (LSAs). The OSPF Update Packet-Pacing Configurable Timers feature allows you to configure the rate at which OSPF LSA flood pacing, retransmission pacing, and group pacing updates occur.

- Configuring OSPF flood pacing timers allows you to control interpacket spacing between consecutive link-state update packets in the OSPF transmission queue.
- Configuring OSPF retransmission pacing timers allows you to control interpacket spacing between consecutive link-state update packets in the OSPF retransmission queue.

- Cisco IOS XE software groups the periodic refresh of LSAs to improve the LSA packing density for the refreshes in large topologies. The group timer controls the interval that is used for group LSA refreshment; however, this timer does not change the frequency at which individual LSAs are refreshed (the default refresh occurs every 30 minutes).

Caution The default settings for OSPF packet-pacing timers are suitable for the majority of OSPF deployments. You should change the default timers only as a last resort.

Benefits of OSPF Update Packet-Pacing Configurable Timers

The OSPF Update Packet-Pacing Configurable Timers feature provides the administrator with a mechanism to control the rate at which LSA updates occur in order to reduce high CPU or buffer utilization that can occur when an area is flooded with a very large number of LSAs.

How to Configure OSPF Packet-Pacing Timers

The tasks in this section describe how to configure and verify three OSPF update packet-pacing timers.

Configuring OSPF Packet-Pacing Timers

Caution The default settings for OSPF packet-pacing timers are suitable for the majority of OSPF deployments. You should change the default timers only as a last resort.

To configure a flood packet-pacing timer, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **router ospf** process-id
2. Router(config-router)# **timers pacing flood** milliseconds

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router ospf** process-id	Places the router in router configuration mode and enables an OSPF routing process.
Step 2	Router(config-router)# **timers pacing flood** milliseconds	Configures a flood packet-pacing timer delay (in milliseconds).

Configuring a Retransmission Packet-Pacing Timer

To configure a retransmission packet-pacing timer, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **router ospf** process-id
2. Router(config-router)# **timers pacing retransmission** milliseconds

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router ospf** process-id	Places the router in router configuration mode and enables an OSPF routing process.
Step 2	Router(config-router)# **timers pacing retransmission** milliseconds	Configures a retransmission packet-pacing timer delay (in milliseconds).

Configuring a Group Packet-Pacing Timer

To configure a group packet-pacing timer, use the following commands beginning in router configuration mode:

SUMMARY STEPS

1. Router(config)# **router ospf** process-id
2. Router(config-router)# **timers pacing lsa-group** seconds

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **router ospf** process-id	Places the router in router configuration mode and enables an OSPF routing process.
Step 2	Router(config-router)# **timers pacing lsa-group** seconds	Configures an LSA group packet-pacing timer delay (in seconds).

Verifying OSPF Packet-Pacing Timers

To verify that OSPF packet pacing has been configured, use the show ip ospf privileged EXEC command. The output of the show ip ospf command will display the type and delay time of the configurable pacing timers (flood, retransmission, group). The following sample output is from the show ip ospf command:

```
Router# show ip ospf
  Routing Process "ospf 1" with ID 10.0.0.1 and Domain ID 10.20.0.1
  Supports only single TOS(TOS0) routes
  Supports opaque LSA
  SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
```

```
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
LSA group pacing timer 100 secs
Interface flood pacing timer 55 msecs
Retransmission pacing timer 100 msecs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 2. 2 normal 0 stub 0 nssa
External flood list length 0
   Area BACKBONE(0)
       Number of interfaces in this area is 2
       Area has message digest authentication
       SPF algorithm executed 4 times
       Area ranges are
       Number of LSA 4. Checksum Sum 0x29BEB
       Number of opaque link LSA 0. Checksum Sum 0x0
       Number of DCbitless LSA 3
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
   Area 172.16.26.0
       Number of interfaces in this area is 0
       Area has no authentication
       SPF algorithm executed 1 times
       Area ranges are
          192.168.0.0/16 Passive Advertise
       Number of LSA 1. Checksum Sum 0x44FD
       Number of opaque link LSA 0. Checksum Sum 0x0
       Number of DCbitless LSA 1
       Number of indication LSA 1
       Number of DoNotAge LSA 0
       Flood list length 0
```

Troubleshooting Tips

If the number of OSPF packet retransmissions rapidly increases, increase the value of the packet-pacing timers. The number of OSPF packet retransmissions is displayed in the output of the show ip ospf neighbor command.

Monitoring and Maintaining OSPF Packet-Pacing Timers

Command	Purpose
`Router# show ip ospf`	Displays general information about OSPF routing processes.
`router# show ip ospf neighbor`	Displays OSPF neighbor information on a per-interface basis.
`Router# clear ip ospf redistribution`	Clears route redistribution based on the OSPF routing process ID.

Configuration Examples of OSPF Update Packet-Pacing

Example LSA Flood Pacing

The following example configures LSA flood pacing updates to occur in 50-millisecond intervals for OSPF routing process 1:

```
Router(config)# router ospf 1
Router(config-router)# timers pacing flood 50
```

Example LSA Retransmission Pacing

The following example configures LSA retransmission pacing updates to occur in 100-millisecond intervals for OSPF routing process 1:

```
Router(config)# router ospf 1
Router(config-router)# timers pacing retransmission 100
```

Example LSA Group Pacing

The following example configures OSPF group pacing updates between LSA groups to occur in 75-second intervals for OSPF routing process 1:

```
Router(config)# router ospf 1
Router(config-router)# timers pacing lsa-group 75
```

Additional References

For additional information related to the OSPF Update Packet-Pacing Configurable Timers feature, see the following references:

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases , and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Update Packet-Pacing Configurable Timers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 220: Feature Information for OSPF Update Packet-Pacing Configurable Timers

Feature Name	Releases	Feature Information
OSPF Update Packet-Pacing Configurable Timers	Cisco IOS XE Release 2.1	The OSPF Update Packet-Pacing Configurable Timers feature allows you to configure the rate at which OSPF LSA flood pacing, retransmission pacing, and group pacing updates occur. The following commands are introduced or modified in the feature documented in this module: • **timers pacing flood** • **timers pacing lsa-group** • **timers pacing retransmission** • **show ip ospf**

Table 221: Feature Information for OSPF Update Packet-Pacing Configurable Timers

Feature Name	Releases	Feature Information
OSPF Update Packet-Pacing Configurable Timers	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 202

OSPF Sham-Link Support for MPLS VPN

This document describes how to configure and use a sham-link to connect Virtual Private Network (VPN) client sites that run the Open Shortest Path First (OSPF) protocol and share backdoor OSPF links in a Multiprotocol Label Switching (MPLS) VPN configuration.

Prerequisites for OSPF Sham-Link Support for MPLS VPN

Before you can configure a sham-link in an MPLS VPN, you must first enable OSPF as follows:

- Create an OSPF routing process.
- Specify the range of IP addresses to be associated with the routing process.
- Assign area IDs to be associated with the range of IP addresses.

Restrictions on OSPF Sham-Link Support for MPLS VPN

When OSPF is used as a protocol between PE and CE routers, the OSPF metric is preserved when routes are advertised over the VPN backbone. The metric is used on the remote PE routers to select the correct route. For this reason, you should not modify the metric value when OSPF is redistributed to BGP, and when BGP is redistributed to OSPF. If you modify the metric value, routing loops may occur.

Information About OSPF Sham-Link Support for MPLS VPN

Benefits of OSPF Sham-Link Support for MPLS VPN

Client Site Connection Across the MPLS VPN Backbone

A sham-link overcomes the OSPF default behavior for selecting an intra-area backdoor route between VPN sites instead of an interarea (PE-to-PE) route. A sham-link ensures that OSPF client sites that share a backdoor link can communicate over the MPLS VPN backbone and participate in VPN services.

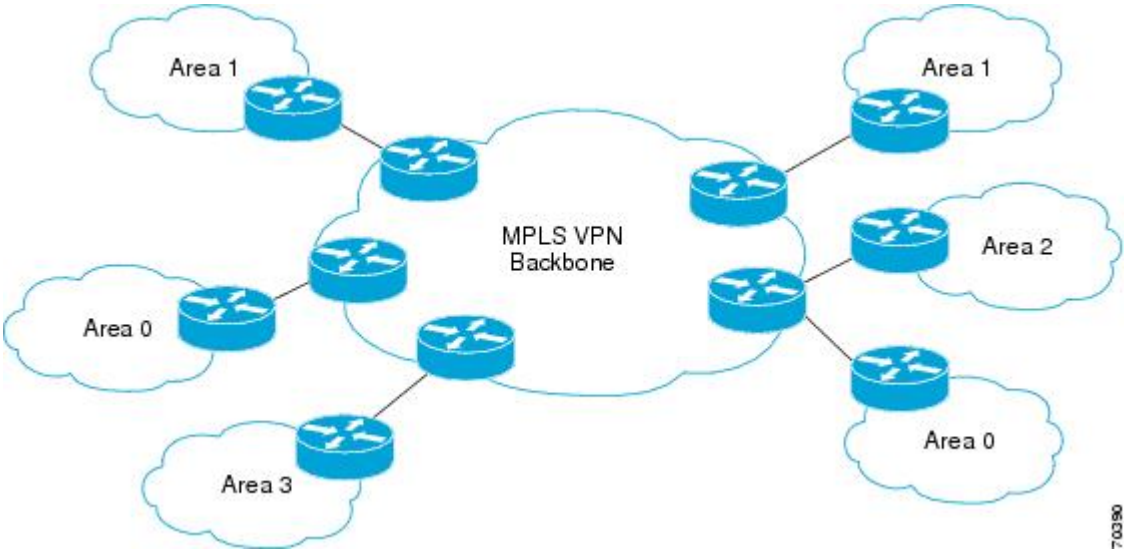
Flexible Routing in an MPLS VPN Configuration

In an MPLS VPN configuration, the OSPF cost configured with a sham-link allows you to decide if OSPF client site traffic will be routed over a backdoor link or through the VPN backbone.

Using OSPF in PE-CE Router Connections

In an MPLS VPN configuration, the OSPF protocol is one way you can connect customer edge (CE) routers to service provider edge (PE) routers in the VPN backbone. OSPF is often used by customers who run OSPF as their intrasite routing protocol, subscribe to a VPN service, and want to exchange routing information between their sites using OSPF (during migration or on a permanent basis) over an MPLS VPN backbone.

The figure below shows an example of how VPN client sites that run OSPF can connect over an MPLS VPN backbone.



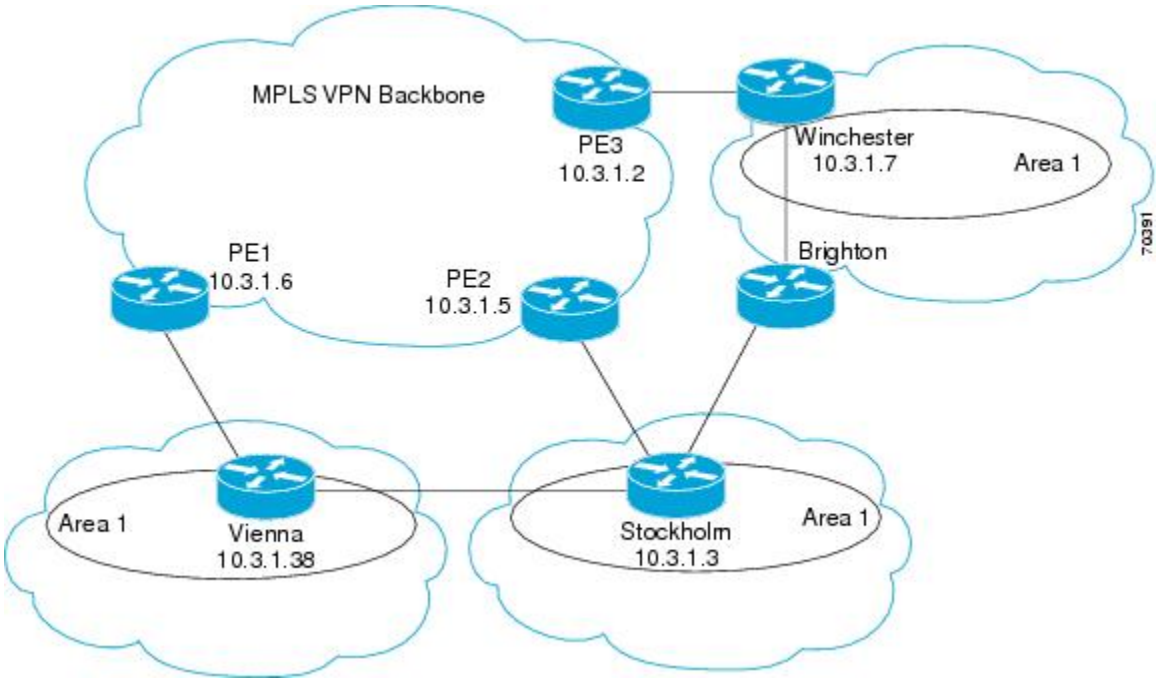
When OSPF is used to connect PE and CE routers, all routing information learned from a VPN site is placed in the VPN routing and forwarding (VRF) instance associated with the incoming interface. The PE routers that attach to the VPN use the Border Gateway Protocol (BGP) to distribute VPN routes to each other. A CE router can then learn the routes to other sites in the VPN by peering with its attached PE router. The MPLS VPN superbackbone provides an additional level of routing hierarchy to interconnect the VPN sites running OSPF.

When OSPF routes are propagated over the MPLS VPN backbone, additional information about the prefix in the form of BGP extended communities (route type, domain ID extended communities) is appended to the BGP update. This community information is used by the receiving PE router to decide the type of link-state advertisement (LSA) to be generated when the BGP route is redistributed to the OSPF PE-CE process. In this way, internal OSPF routes that belong to the same VPN and are advertised over the VPN backbone are seen as interarea routes on the remote sites.

For basic information about how to configure an MPLS VPN, refer to the *Cisco IOS XE MPLS Configuration Guide, Release 2*.

Using a Sham-Link to Correct OSPF Backdoor Routing

Although OSPF PE-CE connections assume that the only path between two client sites is across the MPLS VPN backbone, backdoor paths between VPN sites (shown in grey in the figure below) may exist. If these sites belong to the same OSPF area, the path over a backdoor link will always be selected because OSPF prefers intraarea paths to interarea paths. (PE routers advertise OSPF routes learned over the VPN backbone as interarea paths.) For this reason, OSPF backdoor links between VPN sites must be taken into account so that routing is performed based on policy.



For example, the figure above shows three client sites, each with backdoor links. Because each site runs OSPF within the same Area 1 configuration, all routing between the three sites follows the intraarea path across the backdoor links, rather than over the MPLS VPN backbone.

The following example shows BGP routing table entries for the prefix 10.3.1.7/32 in the PE-1 router in the figure above. This prefix is the loopback interface of the Winchester CE router. As shown in bold in this example, the loopback interface is learned via BGP from PE-2 and PE-3. It is also generated through redistribution into BGP on PE-1.

```
PE-1# show ip bgp vpnv4 all 10.3.1.7
BGP routing table entry for 100:251:10.3.1.7/32, version 58
Paths: (3 available, best #2)
```

```
  Advertised to non peer-group peers:
  10.3.1.2 10.3.1.5
  Local
    10.3.1.5 (metric 30) from 10.3.1.5 (10.3.1.5)
      Origin incomplete, metric 22, localpref 100, valid, internal
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
  Local
    10.2.1.38 from 0.0.0.0 (10.3.1.6)
      Origin incomplete, metric 86, localpref 100, weight 32768,
      valid, sourced, best
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
  Local
    10.3.1.2 (metric 30) from 10.3.1.2 (10.3.1.2)
      Origin incomplete, metric 11, localpref 100, valid, internal
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
```

Within BGP, the locally generated route (10.2.1.38) is considered to be the best route. However, as shown in bold in the next example, the VRF routing table shows that the selected path is learned via OSPF with a next hop of 10.2.1.38, which is the Vienna CE router.

```
PE-1# show ip route vrf ospf 10.3.1.7
Routing entry for 10.3.1.7/32
  Known via "ospf 100", distance 110, metric 86, type intra area
  Redistributing via bgp 215
  Advertised by bgp 215
  Last update from 10.2.1.38 on Serial0/0/0, 00:00:17 ago
  Routing Descriptor Blocks:
  * 10.2.1.38
, from 10.3.1.7, 00:00:17 ago, via Serial0/0/0
      Route metric is 86, traffic share count is 1
```

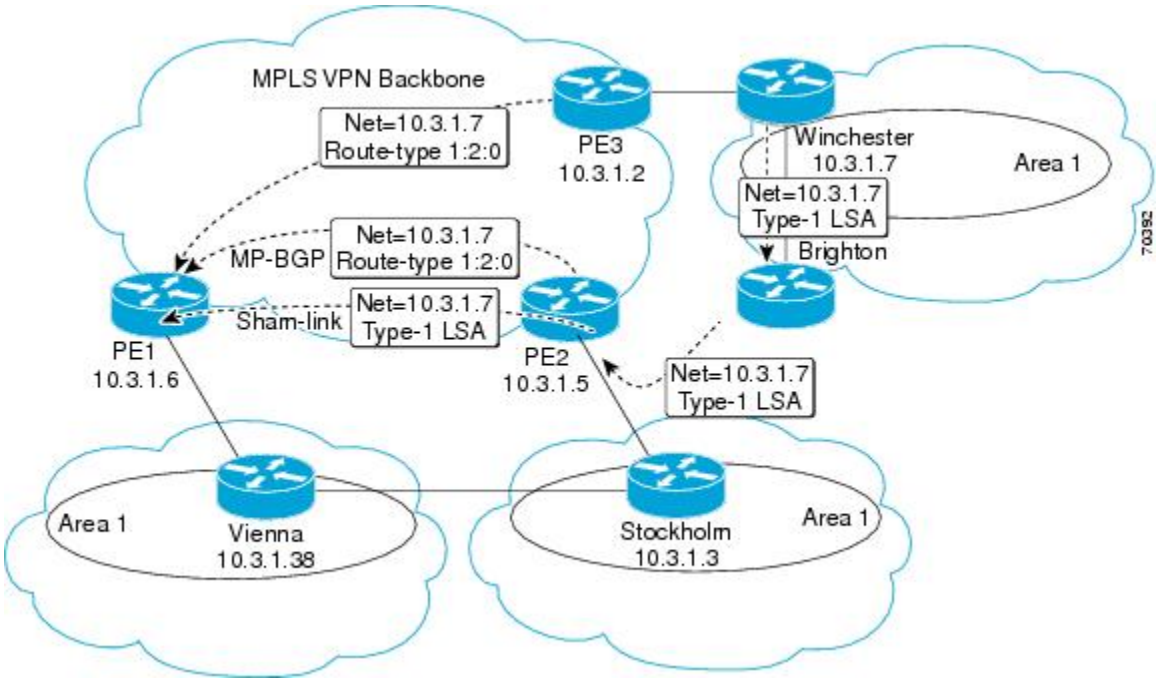
This path is selected because:

- The OSPF intra-area path is preferred over the interarea path (over the MPLS VPN backbone) generated by the PE-1 router.
- OSPF has a lower administrative distance (AD) than internal BGP (BGP running between routers in the same autonomous system).

If the backdoor links between sites are used only for backup purposes and do not participate in the VPN service, then the default route selection shown in the preceding example is not acceptable. To reestablish the desired path selection over the MPLS VPN backbone, you must create an additional OSPF intra-area (logical) link between ingress and egress VRFs on the relevant PE routers. This link is called a sham-link.

A sham-link is required between any two VPN sites that belong to the same OSPF area and share an OSPF backdoor link. If no backdoor link exists between the sites, no sham-link is required.

The figure below shows a sample sham-link between PE-1 and PE-2. A cost is configured with each sham-link and is used to decide whether traffic will be sent over the backdoor path or the sham-link path. When a sham-link is configured between PE routers, the PEs can populate the VRF routing table with the OSPF routes learned over the sham-link.



Because the sham-link is seen as an intra-area link between PE routers, an OSPF adjacency is created and database exchange (for the particular OSPF process) occurs across the link. The PE router can then flood LSAs between sites from across the MPLS VPN backbone. As a result, the desired intra-area connectivity is created.

How to Configure an OSPF Sham-Link

Creating a Sham-Link

Before you begin

Before you create a sham-link between PE routers in an MPLS VPN, you must:

- Configure a separate /32 address on the remote PE so that OSPF packets can be sent over the VPN backbone to the remote end of the sham-link. The /32 address must meet the following criteria:
  - Belong to a VRF.
  - Not be advertised by OSPF.
  - Be advertised by BGP.

You can use the /32 address for other sham-links.

- Associate the sham-link with an existing OSPF area.

To create a sham-link, use the following commands starting in EXEC mode:

SUMMARY STEPS

1. Router1# **configure terminal**
2. Router1(config)# **ip vrf** *vrf-name*
3. Router1(config-vrf)# **exit**
4. Router1(config)# **interface loopback** *interface-number*
5. Router1(config-if)# **ip vrf forwarding** *vrf-name*
6. Router1(config-if)# **ip address** *ip-address mask*
7. Router1(config-if)# **end**
8. Router1(config)# **end**
9. Router2# **configure terminal**
10. Router2(config)# **interface loopback** *interface-number*
11. Router2(config-if)# **ip vrf forwarding** *vrf-name*
12. Router2(config-if)# **ip address** *ip-address mask*
13. Router2(config-if)# **end**
14. Router1(config)# **end**
15. Router1(config)# **router ospf** *process-id* **vrf** *vrf-name*
16. Router1(config-if)# **area** *area-id* **sham-link** *source-address destination-address* **cost** *number*
17. Router2(config)# **router ospf** *process-id* **vrf** *vrf-name*
18. Router2(config-if)# **area** *area-id* **sham-link** *source-address destination-address* **cost** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router1# **configure terminal**	Enters global configuration mode on the first PE router.
Step 2	Router1(config)# **ip vrf** *vrf-name*	Defines a VPN routing and forwarding (VRF) instance and enters VRF configuration mode.
Step 3	Router1(config-vrf)# **exit**	Exits VRF configuration mode and returns to global confiuration mode.
Step 4	Router1(config)# **interface loopback** *interface-number*	Creates a loopback interface to be used as an endpoint of the sham-link on PE-1 and enters interface configuration mode.
Step 5	Router1(config-if)# **ip vrf forwarding** *vrf-name*	Associates the loopback interface with a VRF. Removes the IP address.
Step 6	Router1(config-if)# **ip address** *ip-address mask*	Reconfigures the IP address of the loopback interface on PE-1.
Step 7	Router1(config-if)# **end**	Returns to global configuration mode.
Step 8	Router1(config)# **end**	Returns to EXEC mode.
Step 9	Router2# **configure terminal**	Enters global configuration mode on the second PE router.

	Command or Action	Purpose
Step 10	Router2(config)# **interface loopback** *interface-number*	Creates a loopback interface to be used as the endpoint of the sham-link on PE-2 and enters interface configuration mode.
Step 11	Router2(config-if)# **ip vrf forwarding** *vrf-name*	Associates the second loopback interface with a VRF. Removes the IP address.
Step 12	Router2(config-if)# **ip address** *ip-address mask*	Reconfigures the IP address of the loopback interface on PE-2.
Step 13	Router2(config-if)# **end**	Returns to global configuration mode.
Step 14	Router1(config)# **end**	Returns to EXEC mode.
Step 15	Router1(config)# **router ospf** *process-id* **vrf** *vrf-name*	Configures the specified OSPF process with the VRF associated with the sham-link interface on PE-1 and enters interface configuration mode.
Step 16	Router1(config-if)# **area** *area-id* **sham-link** *source-address destination-address* **cost** *number*	Configures the sham-link on the PE-1 interface within a specified OSPF area and with the loopback interfaces specified by the IP addresses as endpoints. **cost** *number* configures the OSPF cost for sending an IP packet on the PE-1 sham-link interface. **Note** When the BGP route to the sham-link destination address is available in RIB regardless of the source address in RIB, the sham-link is considered up. For example, if shutdown source interface loopback, the sham-link will still be in up state, however it will go in down state after the device reboot.
Step 17	Router2(config)# **router ospf** *process-id* **vrf** *vrf-name*	Configures the specified OSPF process with the VRF associated with the sham-link interface on PE-2 and enters interface configuration mode.
Step 18	Router2(config-if)# **area** *area-id* **sham-link** *source-address destination-address* **cost** *number*	Configures the sham-link on the PE-2 interface within a specified OSPF area and with the loopback interfaces specified by the IP addresses as endpoints. **cost** *number* configures the OSPF cost for sending an IP packet on the PE-2 sham-link interface. **Note** When the BGP route to the sham-link destination address is available in RIB regardless of the source address in RIB, the sham-link is considered up. For example, if shutdown source interface loopback, the sham-link will still be in up state, however it will go in down state after the device reboot.

Verifying Sham-Link Creation

To verify that the sham-link was successfully created and is operational, use the **show ip ospf sham-links** command in EXEC mode:

```
Router# show ip ospf sham-links
Sham Link OSPF_SL0 to address 10.2.1.2 is up
Area 1 source address 10.2.1.1
  Run as demand circuit
  DoNotAge LSA allowed. Cost of using 40 State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Hello due in 00:00:04
    Adjacency State FULL (Hello suppressed)
    Index 2/2, retransmission queue length 4, number of
    retransmission 0
    First 0x63311F3C(205)/0x63311FE4(59) Next
    0x63311F3C(205)/0x63311FE4(59)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
    Link State retransmission due in 360 msec
```

Monitoring and Maintaining a Sham-Link

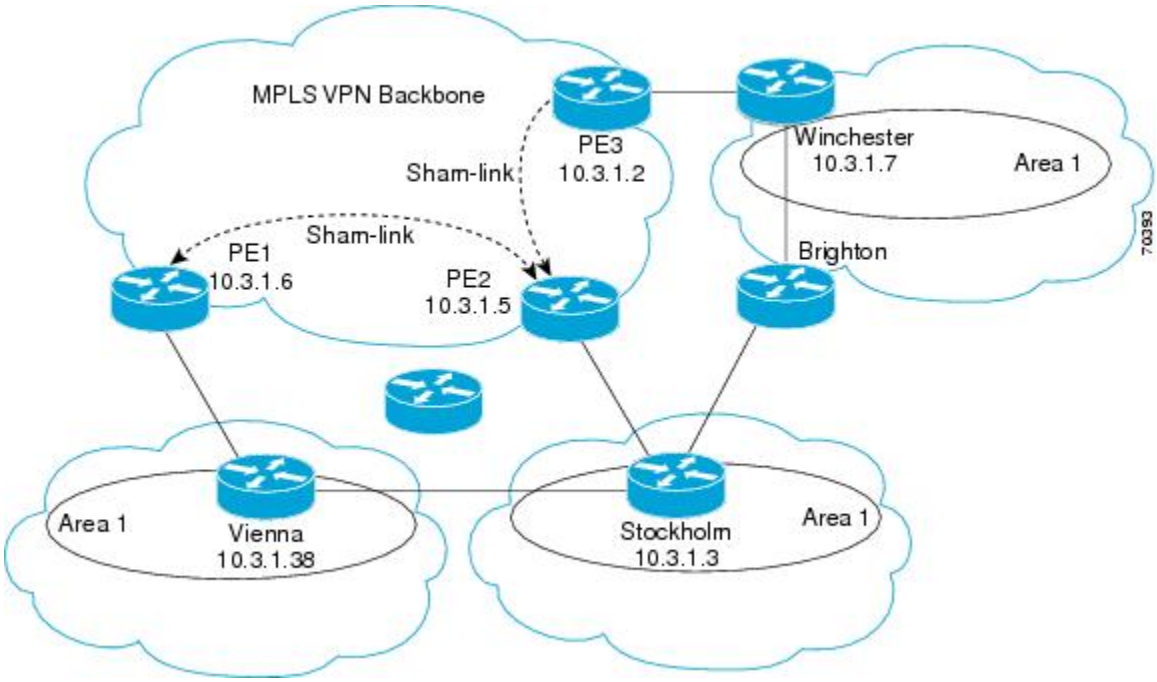
Command	Purpose
`Router# show ip ospf sham-links`	Displays the operational status of all sham-links configured for a router.
`Router# show ip ospf data router` *ip-address*	Displays information about how the sham-link is advertised as an unnumbered point-to-point connection between two PE routers.

Configuration Examples of an OSPF Sham-Link

Example Sham-Link Configuration

This example is designed to show how a sham-link is used only to affect the OSPF intra-area path selection of the PE and CE routers. The PE router also uses the information received from MP-BGP to set the outgoing label stack of incoming packets, and to decide to which egress PE router to label switch the packets.

The figure below shows a sample MPLS VPN topology in which a sham-link configuration is necessary. A VPN client has three sites, each with a backdoor link. Two sham-links have been configured, one between PE-1 and PE-2, and another between PE-2 and PE-3. A sham-link between PE-1 and PE-3 is not necessary in this configuration because the Vienna and Winchester sites do not share a backdoor link.



The following output shows the forwarding that occurs between sites from the standpoint of how PE-1 views the 10.3.1.7/32 prefix, the loopback1 interface of the Winchester CE router in the figure.

```
PE-1# show ip bgp vpnv4 all 10.3.1.7
BGP routing table entry for 100:251:10.3.1.7/32, version 124
Paths: (1 available, best #1)
  Local
    10.3.1.2 (metric 30) from 10.3.1.2
 (10.3.1.2)
      Origin incomplete, metric 11, localpref 100, valid, internal,
      best
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
PE-1# show ip route vrf ospf 10.3.1.7
Routing entry for 10.3.1.7/32
  Known via "ospf 100
", distance 110, metric 13, type intra area
  Redistributing via bgp 215
  Last update from 10.3.1.2 00:12:59 ago
  Routing Descriptor Blocks:
10.3.1.2 (Default-IP-Routing-Table), from 10.3.1.7, 00:12:59 ago
```

The following output shows forwarding information in which the next hop for the route, 10.3.1.2, is the PE-3 router rather than the PE-2 router (which is the best path according to OSPF). The reason the OSPF route is not redistributed to BGP on the PE is because the other end of the sham-link already redistributed the route to BGP and there is no need for duplication. The OSPF sham-link is used only to influence intra-area path selection. When sending traffic to a particular destination, the PE router uses the MP-BGP forwarding information.

```
PE-1# show ip bgp vpnv4 all tag | begin 10.3.1.7
   10.3.1.7/32      10.3.1.2
        notag/38

PE-1# show tag-switching forwarding 10.3.1.2
```

```
Local  Outgoing   Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC  or Tunnel Id     switched   interface
31     42         10.3.1.2/32
       0          PO3/0/0    point2point
PE-1# show ip cef vrf ospf 10.3.1.7
10.3.1.7/32, version 73, epoch 0, cached adjacency to POS3/0/0
0 packets, 0 bytes
  tag information set
   local tag: VPN-route-head
   fast tag rewrite with PO3/0/0, point2point, tags imposed: {42 38
}
  via 10.3.1.2
, 0 dependencies, recursive
    next hop 10.1.1.17, POS3/0/0 via 10.3.1.2/32
    valid cached adjacency
    tag rewrite with PO3/0/0, point2point, tags imposed: {42 38}
```

If a prefix is learned across the sham-link and the path via the sham-link is selected as the best, the PE router does not generate an MP-BGP update for the prefix. It is not possible to route traffic from one sham-link over another sham-link.

In the following output, PE-2 shows how an MP-BGP update for the prefix is not generated. Although 10.3.1.7/32 has been learned via OSPF across the sham-link as shown in bold, no local generation of a route into BGP is performed. The only entry within the BGP table is the MP-BGP update received from PE-3 (the egress PE router for the 10.3.1.7/32 prefix).

```
PE-2# show ip route vrf ospf 10.3.1.7
Routing entry for 10.3.1.7/32
  Known via "ospf 100
", distance 110, metric 12, type intra area
  Redistributing via bgp 215
  Last update from 10.3.1.2 00:00:10 ago
  Routing Descriptor Blocks:
  * 10.3.1.2 (Default-IP-Routing-Table), from 10.3.1.7, 00:00:10 ago
      Route metric is 12, traffic share count is 1
PE-2# show ip bgp vpnv4 all 10.3.1.7
BGP routing table entry for 100:251:10.3.1.7/32, version 166
Paths: (1 available, best #1)
  Not advertised to any peer
  Local
    10.3.1.2 (metric 30) from 10.3.1.2 (10.3.1.2)
      Origin incomplete, metric 11, localpref 100, valid, internal,
      best
      Extended Community: RT:1:793 OSPF DOMAIN ID:0.0.0.100 OSPF
      RT:1:2:0 OSPF 2
```

The PE router uses the information received from MP-BGP to set the ongoing label stack of incoming packets, and to decide to which egress PE router to label switch the packets.

Example Sham-Link Between Two PE Routers

The following example shows how to configure a sham-link between two PE routers:

```
Router1(config)
# interface loopback 1
Router1(config-if)# ip vrf forwarding ospf
Router1(config-if)# ip address 10.2.1.1 255.255.255.255
!
Router2(config)# interface loopback 1
Router2(config-if)# ip vrf forwarding ospf
```

```
Router2(config-if)# ip address 10.2.1.2 255.255.255.255
!
Router1(config)# router ospf 100 vrf ospf
Router1(config-if)# area 1 sham-link 10.2.1.1 10.2.1.2 cost 40
!
Router2(config)# router ospf 100 vrf ospf
Router2(config-if)# area 1 sham-link 10.2.1.2 10.2.1.1 cost 40
```

Additional References

The following sections provide references related to the OSPF Sham-Link Support for MPLS VPN feature.

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases
MPLS Virtual Private Networks	"MPLS Virtual Private Networks"

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1163	*A Border Gateway Protocol*
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 2283	*Multiprotocol Extensions for BGP-4*
RFC 2328	*Open Shortest Path First, Version 2*

RFC	Title
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Sham-Link Support for MPLS VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 222: Feature Information for OSPF Sham-Link Support for MPLS VPN

Feature Name	Releases	Feature Information
OSPF Sham-Link Support for MPLS VPN	Cisco IOS XE Release 2.1	This feature allows you to use a sham-link to connect Virtual Private Network (VPN) client sites that run OSPF and share backdoor OSPF links in a Multiprotocol Label Switching (MPLS) VPN configuration. The following commands are introduced or modified in the feature documented in this module: • **area sham-link cost** • **show ip ospf sham-links**

Table 223: Feature Information for OSPF Sham-Link Support for MPLS VPN

Feature Name	Releases	Feature Information
OSPF Sham-Link Support for MPLS VPN	Cisco IOS XE Release 17.4	This feature was introduced.

Glossary

BGP --Border Gateway Protocol. Interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined in RFC 1163.

CE router --customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers are not aware of associated VPNs.

CEF -- Cisco Express Forwarding. An advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns.

IGP --Interior Gateway Protocol. An Internet protocol used to exchange routing information within an autonomous system. Examples of common IGPs include IGRP, OSPF, and RIP.

LSA --link-state advertisement. A broadcast packet used by link-state protocols. The LSA contains information about neighbors and path costs and is used by the receiving router to maintain a routing table.

MPLS --Multiprotocol Label Switching. Emerging industry standard upon which tag switching is based.

OSPF --Open Shortest Path First protocol.

PE router --provider edge router. A router that is part of a service provider network connected to a customer edge (CE) router. All VPN processing occurs in the PE router.

SPF --shortest path first calculation.

VPN --Virtual Private Network. A secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.

VRF --VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

CHAPTER 203

OSPF Support for Multi-VRF on CE Routers

The OSPF Support for Multi-VRF on CE Routers feature provides the capability to suppress provider edge (PE) checks that are needed to prevent loops when the PE is performing a mutual redistribution of packets between the OSPF and BGP protocols. When VPN routing and forwarding (VRF) is used on a router that is not a PE (that is, one that is not running BGP), the checks can be turned off to allow for correct population of the VRF routing table with routes to IP prefixes.

OSPF multi-VRF allows you to split the router into multiple virtual routers, where each router contains its own set of interfaces, routing table, and forwarding table.

Information About OSPF Support for Multi-VRF on CE Routers

The OSPF Support for Multi-VRF on CE Routers feature provides the capability to suppress provider edge (PE) checks that are needed to prevent loops when the PE is performing a mutual redistribution of packets between the OSPF and BGP protocols. When VPN routing and forwarding (VRF) is used on a router that is not a PE (that is, one that is not running BGP), the checks can be turned off to allow for correct population of the VRF routing table with routes to IP prefixes.

OSPF multi-VRF allows you to split the router into multiple virtual routers, where each router contains its own set of interfaces, routing table, and forwarding table. OSPF multi-VRF gives you the ability to segment parts of your network and configure those segments to perform specific functions, yet still maintain correct routing information.

How to Configure OSPF Support for Multi-VRF on CE Routers

Configuring the Multi-VRF Capability for OSPF Routing

Before you begin

CEF must be running on the network.

SUMMARY STEPS

1. **enable**
2. **show ip ospf** [*process-id*
3. **configure terminal**
4. **vpdn- group** *name*
5. **exit**
6. **resource-pool profile vpdn** *name*
7. **vpdn group** *name*
8. **vpn vrf** *vrf-name* | **id** *vpn-id*
9. **exit**
10. **router ospf** *process-id* [**vrf** *vpn-name*]
11. **capability vrf-lite**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip ospf** [*process-id* **Example:** `Router# show ip ospf 1`	Displays the status of the router. If the display indicates that the router is connected to the VPN backbone, you can use the **capability vrf-lite** command to decouple the PE router from the VPN backbone.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**vpdn- group** *name* **Example:** `Router(config)# vpdn-group mygroup`	Creates a VPDN group.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-vpdn)# exit`	Leaves the configuration mode and returns to global configuration mode.
Step 6	**resource-pool profile vpdn** *name* **Example:** `Router(config)# resource-pool profile vpdn company1`	Creates a virtual private dialup network (VPDN) profile and enters VPDN profile configuration mode.
Step 7	**vpdn group** *name* **Example:** `Router(config-vpdn-profile)# vpdn group mygroup`	Associates a virtual private dialup network (VPDN) group with a customer or VPDN profile.
Step 8	**vpn vrf** *vrf-name* | **id** *vpn-id* **Example:** `Router(config-vpdn)# vpn vrf grc`	Specifies that the source and destination IPv4 addresses of a given virtual private dialup network (VPDN) group belong to a specified Virtual Private Network (VPN) routing and forwarding (VRF) instance.
Step 9	**exit** **Example:** `Router(config-vpdn)# exit`	Leaves the configuration mode and returns to global configuration mode.
Step 10	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Router(config)# router ospf 1 vrf grc`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process. • Use the **vrf** keyword and *vpn-name* argument to identify a VPN.
Step 11	**capability vrf-lite** **Example:** `Router(config-router)# capability vrf-lite`	Applies the multi-VRF capability to the OSPF process.

Verifying the OSPF Multi-VRF Configuration

No specific **debug** or **show** commands are associated with this feature. You can verify the success of the OSPF multi-VRF configuration by using the **show ip ospf** *process-id*] command to verify that the router is not connected to the VPN backbone.

This output from the **show ip ospf process**command indicates that the PE router is currently connected to the backbone.

```
Router# show ip ospf 12
 Routing Process "ospf 12" with ID 172.16.1.1 and Domain ID 0.0.0.12
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Connected to MPLS VPN Superbackbone
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x0
 Number of opaque AS LSA 0. Checksum Sum 0x0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0
```

When the OSPF VRF process is configured with the **capability vrf-lite** command under the **router ospf** command, the "Connected to MPLS VPN Superbackbone" line will not be present in the display.

Configuration Example for OSPF Support for Multi-VRF on CE Routers

Example Configuring the Multi-VRF Capability

This example shows a basic OSPF network with a VRF named grc configured. The **capability vrf-lite** command is entered to suppress the PE checks.

```
!
ip cef
ip vrf grc
 rd 1:1
interface Serial2/0/0
 ip vrf forwarding grc
 ip address 192.168.1.1 255.255.255.252
!
interface Serial3/0/0
 ip vrf forwarding grc
 ip address 192.168.2.1 255.255.255.252
...
!
router ospf 9000 vrf grc
 log-adjacency-changes
 capability vrf-lite
 redistribute rip metric 1 subnets
 network 192.168.1.0 0.0.0.255 area 0
!
router rip
 address-family ipv4 vrf grc
 redistribute ospf 9000 vrf grc
 network 192.168.2.0
 no auto-summary
 end
Device# show ip route vrf grc
Routing Table: grc
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
O IA 192.168.192.0/24 [110/138] via 192.168.1.13, 00:06:08, Serial2/0/0
                      [110/138] via 192.168.1.9, 00:06:08, Serial3/0/0
O IA 192.168.242.0/24 [110/74] via 192.168.1.13, 00:06:08, Serial2/0/0
O IA 192.168.193.0/24 [110/148] via 192.168.1.13, 00:06:08, Serial2/0/0
                      [110/148] via 192.168.1.9, 00:06:08, Serial3/0/0
O IA 192.168.128.0/24 [110/74] via 192.168.1.9, 00:06:08, Serial3/0/0
O IA 192.168.129.0/24 [110/84] via 192.168.1.9, 00:06:08, Serial3/0/0
O IA 192.168.130.0/24 [110/84] via 192.168.1.9, 00:06:08, Serial3/0/0
     172.16.0.0/24 is subnetted, 2 subnets
O E2    172.16.9.0 [110/5] via 192.168.1.13, 00:06:08, Serial2/0/0
O E2    172.16.10.0 [110/5] via 192.168.1.13, 00:06:08, Serial2/0/0
O IA 192.168.131.0/24 [110/94] via 192.168.1.9, 00:06:20, Serial3/0/0
     192.168.1.0/30 is subnetted, 4 subnets
C       192.168.1.8 is directly connected, Serial3/0/0
C       192.168.1.12 is directly connected, Serial2/0/0
O       192.168.1.0 [110/128] via 192.168.1.9, 00:06:20, Serial3/0/0
O       192.168.1.4 [110/128] via 192.168.1.13, 00:06:20, Serial2/0/0
```

Additional References

For additional information related to OSPF support for multi-VRF on CE routers, see the following references.

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
Multiprotocol Label Switching (MPLS)	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide, Release 2*
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases , and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Support for Multi-VRF on CE Routers

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 224: Feature Information for OSPF Support for Multi-VRF on CE Routers

Feature Name	Releases	Feature Information
OSPF Support for Multi-VRF on CE Routers	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.1.0 SG	The OSPF Support for Multi-VRF on CE Routers feature provides the capability to suppress provider edge (PE) checks that are needed to prevent loops when the PE is performing a mutual redistribution of packets between the OSPF and BGP protocols. When VPN routing and forwarding (VRF) is used on a router that is not a PE (that is, one that is not running BGP), the checks can be turned off to allow for correct population of the VRF routing table with routes to IP prefixes. The following commands are introduced or modified in the feature documented in this module: • **capability vrf-lite**

Table 225: Feature Information for OSPF Support for Multi-VRF on CE Routers

Feature Name	Releases	Feature Information
OSPF Support for Multi-VRF on CE Routers	Cisco IOS XE Release 17.4	This feature was introduced.

Glossary

CE Router --Customer Edge router, an edge router in the C network, defined as a C router which attaches directly to a P router.

C Network --Customer (enterprise or service provider) network.

C Router --Customer router, a router in the C network.

LSA --link-state advertisement . Broadcast packet used by link-state protocols that contains information about neighbors and path costs. LSAs are used by the receiving routers to maintain their routing tables.

PE Router --Provider Edge router, an edge router in the P network, defined as a P router which attaches directly to a C router.

P Network --MPLS-capable service provider core network. P routers perform MPLS.

P Router --Provider router, a router in the P network.

SPF --shortest path first. A routing algorithm that iterates on length of path to determine a shortest-path spanning tree.

VPN --Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another.

VRF --VPN Routing and Forwarding.

CHAPTER 204

OSPFv3 Multiarea Adjacency

The OSPFv3 Multiarea Adjacency feature allows you to configure a link that multiple Open Shortest Path First version 3 (OSPFv3) areas can share to enable optimized routing. You can add more than one area to an existing OSPFv3 primary interface.

Restrictions for OSPFv3 Multiarea Adjacency

- A multiarea interface operates only if OSPFv3 is configured on the primary interface and the OSPFv3 network type of the primary interface is point-to-point.
- A multiarea interface exists as a logical construct over a primary interface for OSPFv3; however, the neighbor state on the primary interface is independent of the multiarea interface.
- A multiarea interface establishes a neighbor relationship with the corresponding multiarea interface on the neighboring device. A mixture of multiarea and primary interfaces is not supported.
- A multiarea interface advertises a point-to-point connection to another device in the device link-state advertisement (LSA) for the corresponding area when the neighbor state is full.
- A multiarea interface inherits all the OSPFv3 parameters (such as, authentication) from the primary interface. You cannot configure the parameters on a multiarea interface; however, you can configure the parameters on the primary interface.

Information About OSPFv3 Multiarea Adjacency

OSPFv3 Multiarea Adjacency Overview

Open Shortest Path First version 3 (OSPFv3) allows a single physical link to be shared by multiple areas. This creates an intra-area path in each of the corresponding areas sharing the same link. All areas have an

interface on which you can configure OSPFv3. One of these interfaces is designated as the primary interface and others as secondary interfaces.

The OSPFv3 Multiarea Adjacency feature allows you to configure a link on the primary interface to enable optimized routing in multiple areas. Each multiarea interface is announced as a point-to-point unnumbered link. The multiarea interface exists as a logical construct over an existing primary interface. The neighbor state on the primary interface is independent of the neighbor state of the multiarea interface. The multiarea interface establishes a neighbor relationship with the corresponding multiarea interface on the neighboring device. You can only configure multiarea adjacency on an interface that has two OSPFv3 speakers.

Use the **ospfv3 multi-area** command to configure multiarea adjacency on the primary OSPFv3 interface.

How to Configure OSPFv3 Multiarea Adjacency

Configuring OSPFv3 Multiarea Adjacency

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 enable**
5. **ospfv3 multi-area** *multi-area-id*
6. **ospfv3 multi-area** *multi-area-id* **cost** *interface-cost*
7. **ospfv3** *process-id* **ipv6 area** *area-id*
8. **serial restart-delay** *count*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial 2/0`	Specifies the interface type and number.
Step 4	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

	Command or Action	Purpose
Step 5	**ospfv3 multi-area** *multi-area-id* **Example:** `Device(config-if)# ospfv3 multi-area 100`	Configures multiarea adjacency on the interface. • The *multi-area-id* argument identifies the OSPFv3 multiarea. The range is from 0 to 4294967295, or you can use an IP address.
Step 6	**ospfv3 multi-area** *multi-area-id* **cost** *interface-cost* **Example:** `Device(config-if)# ospfv3 multi-area 100 cost 512`	(Optional) Specifies the cost of sending a packet on an OSPFv3 multiarea interface. Use this command to specify the cost only if you want the cost of the multiarea interface to be different than the cost of the primary interface.
Step 7	**ospfv3** *process-id* **ipv6 area** *area-id* **Example:** `Device(config-if)# ospfv3 1 ipv6 area 0`	Configures the OSPFv3 interface. • The *process-id* argument identifies the OSPF process. The range is from 1 to 65535. • The *area-id* argument identifies the OSPF area. The range is from 0 to 4294967295, or you can use an IP address.
Step 8	**serial restart-delay** *count* **Example:** `Device(config-if)# serial restart-delay 0`	Sets the amount of time that the router waits before trying to bring up a serial interface when it goes down. The *count* argument specifies the frequency (in seconds) at which that hardware is reset. The range is from 0 to 900.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying OSPFv3 Multiarea Adjacency

SUMMARY STEPS

1. **enable**
2. **show ospfv3 interface brief**
3. **show ospfv3 multi-area**
4. **show ospfv3 interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ospfv3 interface brief** **Example:**	Displays brief information about Open Shortest Path First version 3 (OSPFv3) interfaces.

	Command or Action	Purpose
	`Device# show ospfv3 interface brief`	
Step 3	**show ospfv3 multi-area** **Example:** `Device# show ospfv3 multi-area`	Displays information about OSPFv3 multiarea interfaces.
Step 4	**show ospfv3 interface** **Example:** `Device# show ospfv3 interface`	Displays information about OSPFv3 interfaces.

Configuration Examples for OSPFv3 Multiarea Adjacency

Example: OSPFv3 Multiarea Adjacency Configuration

```
Device> enable
Device# configure terminal
Device(config)# interface serial 2/0
Device(config-if)# ipv6 enable
Device(config-if)# ospfv3 multi-area 100
Device(config-if)# ospfv3 multi-area 100 cost 512
Device(config-if)# ospfv3 1 ipv6 area 0
Device(config-if)# serial restart-delay 0
Device(config-if)# end
```

Example: Verifying OSPFv3 Multiarea Adjacency

Sample Output for the show ospfv3 interface brief Command

To display brief information about Open Shortest Path First version 3 (OSPFv3) interfaces, use the **show ospfv3 interface brief** command in privileged EXEC mode.

```
Device# show ospfv3 interface brief

Interface PID Area  AF   Cost  State Nbrs F/C
Se2/0     1   0    ipv6  64    P2P   1/1
MA2 1     1   100  ipv6  512   P2P   1/1
```

Sample Output for the show ospfv3 multi-area Command

To display information about OSPFv3 multiarea interfaces, use the **show ospfv3 multi-area** command in privileged EXEC mode.

```
Device# show ospfv3 multi-area

OSPFV3_MA2 is up, line protocol is up
Primary Interface Serial2/0, Area 100
Interface ID 10
```

```
MTU is 1500 bytes
Neighbor Count is 1
```

Sample Output for the show ospfv3 interface Command

To display information about OSPFv3 interfaces, use the **show ospfv3 interface** command in privileged EXEC mode.

```
Device# show ospfv3 interface

Serial2/0 is up, line protocol is up
Link Local Address 2001:DB8:0:ABCD::1, Interface ID 10
Area 0, Process ID 1, Instance ID 0, Router ID 10.0.0.12
Network Type POINT_TO_POINT, Cost: 64
Transmit Delay is 1 sec, State POINT_TO_POINT
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:07
Graceful restart helper support enabled
Index 1/1/1, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 10.0.0.22
Suppress hello for 0 neighbor(s)
Multi-area interface Count is 1
OSPFV3_MA2 interface exists in area 100 Neighbor Count is 1
OSPFV3_MA2 is up, line protocol is up
Link Local Address 2001:DB8:0:ABCD::1, Interface ID 10
Area 100, Process ID 1, Instance ID 0, Router ID 10.0.0.12
Network Type POINT_TO_POINT, Cost: 512
Transmit Delay is 1 sec, State POINT_TO_POINT
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:08
Graceful restart helper support enabled
Index 1/1/2, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 10.0.0.22
```

Additional References for OSPFv3 Multiarea Adjacency

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for OSPFv3 Multiarea Adjacency

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
OSPFv3 Multiarea Adjacency	Cisco IOS XE Release 3.11S	The OSPFv3 Multiarea Adjacency feature allows you to configure a link that multiple Open Shortest Path First version 3 (OSPFv3) areas can share to enable optimized routing. You can add more than one area to an existing OSPFv3 primary interface.

Feature Name	Releases	Feature Information
OSPFv3 Multiarea Adjacency	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 205

OSPFv2 Autoroute Exclude

The OSPFv2 Autoroute Exclude feature allows specific destinations and prefixes to avoid Traffic Engineering (TE) tunnels for the packet transport. The rest of the prefixes can still be set to use TE tunnels. Prefixes that are excluded do not use a TE tunnel path. Only native non-TE paths are downloaded to RIB for such routes. This module describes how to configure the OSPFv2 Autoroute Exclude feature.

Prerequisites for OSPFv2 Autoroute Exclude

- Open Shortest Path First (OSPF) must be configured in your network.
- Cisco Express Forwarding (CEF) must be enabled.
- Multiprotocol Label Switching (MPLS) TE tunnels must be configured.

Information About OSPFv2 Autoroute Exclude

Overview of OSPFv2 Autoroute Exclude

The Autoroute feature is an IP routing method that forces OSPF to use MPLS TE tunnels to build paths for IP traffic routes.

The Autoroute feature enables all routes to use TE Tunnels, even if there is an alternate non-TE path available for that route.

The OSPFv2 Autoroute Exclude feature allows specific destinations or prefixes to avoid TE tunnels, while other prefixes can still be configured to use TE tunnels. Prefixes that are excluded do not use a TE tunnel path. Only native non-TE paths are downloaded to RIB for such routes.

The auto route exclude option is configured under the router OSPF configuration mode by using a prefix list. IP addresses and prefixes that are members of this prefix list are excluded from TE tunnels, even when the auto route is enabled on them. If the IP addresses or prefixes are added to the prefix list, they are dynamically

routed without passing through the TE tunnel. If the IP addresses or prefixes are removed from the prefix list, they are dynamically rerouted back on the TE tunnel path.

How to Configure OSPFv2 Autoroute Exclude

Configuring OSPFv2 Autoroute Exclude

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-ID*
4. **router-id** *ip-address*
5. **mpls traffic-eng router-id** *interface-name*
6. **mpls traffic-eng area***number*
7. **mpls traffic-eng autoroute-exclude prefix-list** *prefix-list-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-ID* **Example:** `Device(config)# router ospf 18`	Configures OSPF routing process and enters OSPF router configuration mode.
Step 4	**router-id** *ip-address* **Example:** `Device(config-router)# router-id 10.1.1.1`	Enables to use a fixed router ID in router configuration mode.
Step 5	**mpls traffic-eng router-id** *interface-name* **Example:** `Device(config-router)# mpls traffic-eng router-id Loopback0`	Specifies the traffic engineering router identifier for the node and the IP address associated with a given interface.
Step 6	**mpls traffic-eng area***number* **Example:**	Configures a router running OSPF MPLS so that it floods traffic engineering for the indicated OSPF area.

	Command or Action	Purpose
	`Device(config-router)# mpls traffic-eng area 0`	
Step 7	**mpls traffic-eng autoroute-exclude prefix-list** *prefix-list-name* **Example:** `Device(config-router)# mpls traffic-eng autoroute-exclude prefix-list kmd`	Allows specific destinations and prefixes to avoid routing through TE tunnels. • Prefixes that are excluded do not use a TE tunnel path.
Step 8	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to privileged EXEC mode.

Configuration Examples for OSPFv2 Autoroute Exclude

Example: Configuring OSPFv2 Autoroute Exclude

```
!
router ospf 1
 router-id 3.3.3.3
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng autoroute-exclude prefix-list XX

!
```

Additional References for OSPFv2 Autoroute Exclude

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Configuring OSPF	*IP Routing: OSPF Configuration Guide*
Configuring Basic Cisco Express Forwarding	*IP Switching: Cisco Express Forwarding Configuration Guide*
MPLS Traffic Engineering Tunnel Source	*MPLS Traffic Engineering Path Calculation and Setup Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for OSPFv2 Autoroute Exclude

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 226: Feature Information for OSPFv2 Autoroute Exclude

Feature Name	Releases	Feature Information
OSPFv2 Autoroute Exclude	Cisco IOS XE 3.13S	The OSPFv2 Autoroute Exclude feature allows specific destinations and prefixes to avoid TE tunnels for the packet transport. The following commands were introduced or modified: **mpls traffic-eng autoroute-exclude prefix list**.

Table 227: Feature Information for OSPFv2 Autoroute Exclude

Feature Name	Releases	Feature Information
OSPFv2 Autoroute Exclude	Cisco IOS XE 17.4	This feature was introduced.

CHAPTER 206

OSPFv3 Address Families

The Open Shortest Path First version 3 (OSPFv3) address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, users may have two processes per interface, but only one process per address family (AF).

Prerequisites for OSPFv3 Address Families

- To use the IPv4 unicast address families (AF) in OSPFv3, you must enable IPv6 on a link, although the link may not be participating in IPv6 unicast AF.
- With the OSPFv3 Address Families feature, users may have two processes per interface, but only one process per AF. If the AF is IPv4, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface.

Information About OSPFv3 Address Families

OSPFv3 Address Families

The OSPFv3 address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, users may have two processes per interface, but only one process per AF. If the IPv4 AF is used, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface. A single IPv4 or IPv6 OSPFv3 process running multiple instances on the same interface is not supported.

Users with an IPv6 network that uses OSPFv3 as its IGP may want to use the same IGP to help carry and install IPv4 routes. All devices on this network have an IPv6 forwarding stack. Some (or all) of the links on this network may be allowed to do IPv4 forwarding and be configured with IPv4 addresses. Pockets of IPv4-only devices exist around the edges running an IPv4 static or dynamic routing protocol. In this scenario,

users need the ability to forward IPv4 traffic between these pockets without tunneling overhead, which means that any IPv4 transit device has both IPv4 and IPv6 forwarding stacks (e.g., is dual stack).

This feature allows a separate (possibly incongruent) topology to be constructed for the IPv4 AF. It installs IPv4 routes in IPv4 RIB, and then the forwarding occurs natively. The OSPFv3 process fully supports an IPv4 AF topology and can redistribute routes from and into any other IPv4 routing protocol.

An OSPFv3 process can be configured to be either IPv4 or IPv6. The **address-family** command is used to determine which AF will run in the OSPFv3 process, and only one address family can be configured per instance. Once the AF is selected, users can enable multiple instances on a link and enable address-family-specific commands.

Different instance ID ranges are used for each AF. Each AF establishes different adjacencies, has a different link state database, and computes a different shortest path tree. The AF then installs the routes in AF-specific RIB. LSAs that carry IPv6 unicast prefixes are used without any modification in different instances to carry each AFs' prefixes.

The IPv4 subnets configured on OSPFv3-enabled interfaces are advertised through intra-area prefix LSAs, just as any IPv6 prefixes. External LSAs are used to advertise IPv4 routes redistributed from any IPv4 routing protocol, including connected and static. The IPv4 OSPFv3 process runs the SPF calculations and finds the shortest path to those IPv4 destinations. These computed routes are then inserted in the IPv4 RIB (computed routes are inserted into an IPv6 RIB for an IPv6 AF).

Because the IPv4 OSPFv3 process allocates a unique pdbindex in the IPv4 RIB, all other IPv4 routing protocols can redistribute routes from it. The parse chain for all protocols is same, so the **ospfv3** keyword added to the list of IPv4 routing protocols causes OSPFv3 to appear in the **redistribute** command from any IPv4 routing protocol. With the **ospfv3** keyword, IPv4 OSPFv3 routes can be redistributed into any other IPv4 routing protocol as defined in the **redistribute ospfv3** command.

Third-party devices will not neighbor with devices running the AF feature for the IPv4 AF because they do not set the AF bit. Therefore, those devices will not participate in the IPv4 AF SPF calculations and will not install the IPv4 OSPFv3 routes in the IPv6 RIB.

How to Configure OSPFv3 Address Families

Configuring the OSPFv3 Router Process

Once you have completed step 3 and entered OSPFv3 router configuration mode, you can perform any of the subsequent steps in this task as needed to perform OSPFv3 device configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **area** *area-ID* [**default-cost** | **nssa** | **stub**]
5. **auto-cost reference-bandwidth** *Mbps*
6. **bfd all-interfaces** [**strict-mode**]
7. **default** {**area** *area-ID*[**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*]

8. **ignore lsa mospf**
9. **interface-id snmp-if-index**
10. **log-adjacency-changes** [**detail**]
11. passive-interface [**default** | *interface-type interface-number*]
12. **queue-depth** {**hello** | **update**} {*queue-size* | **unlimited**}
13. router-id {*router-id*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**area** *area-ID* [**default-cost** | **nssa** | **stub**] **Example:** `Device(config-router)# area 1`	Configures the OSPFv3 area.
Step 5	**auto-cost reference-bandwidth** *Mbps* **Example:** `Device(config-router)# auto-cost reference-bandwidth 1000`	Controls the reference value OSPFv3 uses when calculating metrics for interfaces in an IPv4 OSPFv3 process.
Step 6	**bfd all-interfaces** [**strict-mode**] **Example:** `Device(config-router)# bfd all-interfaces`	Enables BFD for an OSPFv3 routing process **[strict-mode]** - BFD session is established in the strict-mode. In the strict-mode, the OSPF session is not established till the BFD session is established.
Step 7	**default** {**area** *area-ID*[**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*] **Example:**	Returns an OSPFv3 parameter to its default value.

	Command or Action	Purpose
	`Device(config-router)# default area 1`	
Step 8	**ignore lsa mospf** **Example:** `Device(config-router)# ignore lsa mospf`	Suppresses the sending of syslog messages when the device receives LSA Type 6 multicast OSPFv3 packets, which are unsupported.
Step 9	**interface-id snmp-if-index** **Example:** `Device(config-router)# interface-id snmp-if-index`	Configures OSPFv3 interfaces with Simple Network Management Protocol (SNMP) MIB-II interface Index (ifIndex) identification numbers in IPv4 and IPv6.
Step 10	**log-adjacency-changes** [**detail**] **Example:** `Device(config-router)# log-adjacency-changes`	Configures the router to send a syslog message when an OSPFv3 neighbor goes up or down.
Step 11	passive-interface [**default** | *interface-type interface-number*] **Example:** `Device(config-router)# passive-interface default`	Suppresses sending routing updates on an interface when using an IPv4 OSPFv3 process.
Step 12	**queue-depth** {**hello** | **update**} {*queue-size* | **unlimited**} **Example:** `Device(config-router)# queue-depth update 1500`	Configures the number of incoming packets that the IPv4 OSPFv3 process can keep in its queue.
Step 13	router-id {*router-id*} **Example:** `Device(config-router)# router-id 10.1.1.1`	Use a fixed device ID.

Configuring the IPv6 Address Family in OSPFv3

Perform this task to configure the IPv6 address family in OSPFv3. Once you have completed step 4 and entered IPv6 address-family configuration mode, you can perform any of the subsequent steps in this task as needed to configure the IPv6 AF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **address-family ipv6 unicast**

5. **area** *area-ID* **range** *ipv6-prefix* / *prefix-length*
6. **default** {**area** *area-ID*[**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*]
7. **default-information originate** [**always** | **metric** *metric-value* | **metric-type** *type-value*| **route-map** *map-name*]
8. default-metric *metric-value*
9. **distance** *distance*
10. **distribute-list prefix-list** *list-name* {**in**[*interface-type interface-number*] | **out** *routing-process* [*as-number*]}
11. **maximum-paths** *number-paths*
12. **summary-prefix** *prefix* [**not-advertise** | **tag** *tag-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Router(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.

Step 4	**address-family ipv6 unicast**
	Example:
	Example:
	`or`
	Example:
	address-family ipv4
	unicast
	Example:
	`Router(config-router)# address-family ipv6 unicast`
		Enters IPv6 address family configuration mode for OSPFv3.
		or
		Enters IPv4 address family configuration mode for OSPFv3.

	Command or Action	Purpose
	Example: **Example:** or **Example:** `Router(config-router)# address-family ipv4 unicast`	
Step 5	**area** *area-ID* **range** *ipv6-prefix* / *prefix-length* **Example:** `Router(config-router-af)# area 1 range 2001:DB8:0:0::0/128`	Configures OSPFv3 area parameters.
Step 6	**default** {**area** *area-ID*[**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*] **Example:** `Router(config-router-af)# default area 1`	Returns an OSPFv3 parameter to its default value.
Step 7	**default-information originate** [**always** | **metric** *metric-value* | **metric-type** *type-value*| **route-map** *map-name*] **Example:** `Router(config-router-af)# default-information originate always metric 100 metric-type 2`	Generates a default external route into an OSPFv3 for a routing domain.
Step 8	default-metric *metric-value* **Example:** `Router(config-router-af)# default-metric 10`	Sets default metric values for IPv4 and IPv6 routes redistributed into the OSPFv3 routing protocol.
Step 9	**distance** *distance* **Example:** `Router(config-router-af)# distance 200`	Configures an administrative distance for OSPFv3 routes inserted into the routing table.
Step 10	**distribute-list prefix-list** *list-name* {**in**[*interface-type interface-number*] | **out** *routing-process* [*as-number*]} **Example:**	Applies a prefix list to OSPFv3 routing updates that are received or sent on an interface.

	Command or Action	Purpose
	`Router(config-router-af)# distribute-list prefix-list PL1 in Ethernet0/0`	
Step 11	**maximum-paths** *number-paths* **Example:** `Router(config-router-af)# maximum-paths 4`	Controls the maximum number of equal-cost routes that a process for OSPFv3 routing can support.
Step 12	**summary-prefix** *prefix* [**not-advertise** | **tag** *tag-value*] **Example:** `Router(config-router-af)# summary-prefix FEC0::/24`	Configures an IPv6 summary prefix in OSPFv3.

Configuring the IPv4 Address Family in OSPFv3

Perform this task to configure the IPv4 address family in OSPFv3. Once you have completed step 4 and entered IPv4 address-family configuration mode, you can perform any of the subsequent steps in this task as needed to configure the IPv4 AF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **address-family ipv4 unicast**
5. **area** *area-id* **range** *ip-address ip-address-mask* [**advertise** | **not-advertise**] [**cost** *cost*]
6. **default** {**area** *area-ID*[**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*]
7. **default-information originate** [**always** | **metric** *metric-value* | **metric-type** *type-value*| **route-map** *map-name*]
8. default-metric *metric-value*
9. **distance** *distance*
10. **distribute-list prefix-list** *list-name* {**in**[*interface-type interface-number*] | **out** *routing-process* [*as-number*]}
11. **maximum-paths** *number-paths*
12. **summary-prefix** *prefix* [**not-advertise** | **tag** *tag-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**address-family ipv4 unicast** **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters IPv4 address family configuration mode for OSPFv3.
Step 5	**area** *area-id* **range** *ip-address ip-address-mask* [**advertise** | **not-advertise**] [**cost** *cost*] **Example:** `Device(config-router-af)# area 0 range 192.168.110.0 255.255.0.0`	Consolidates and summarizes routes at an area boundary.
Step 6	**default** {**area** *area-ID*[**range** *ipv6-prefix* | **virtual-link** *router-id*]} [**default-information originate** [**always** | **metric** | **metric-type** | **route-map**] | **distance** | **distribute-list** *prefix-list prefix-list-name* {**in** | **out**} [*interface*] | **maximum-paths** *paths* | **redistribute** *protocol* | **summary-prefix** *ipv6-prefix*] **Example:** `Device(config-router-af)# default area 1`	Returns an OSPFv3 parameter to its default value.
Step 7	**default-information originate** [**always** | **metric** *metric-value* | **metric-type** *type-value*| **route-map** *map-name*] **Example:** `Device(config-router-af)# default-information originate always metric 100 metric-type 2`	Generates a default external route into an OSPFv3 for a routing domain.
Step 8	default-metric *metric-value* **Example:** `Device(config-router-af)# default-metric 10`	Sets default metric values for IPv4 and IPv6 routes redistributed into the OSPFv3 routing protocol.

	Command or Action	Purpose
Step 9	**distance** *distance* **Example:** `Device(config-router-af)# distance 200`	Configures an administrative distance for OSPFv3 routes inserted into the routing table.
Step 10	**distribute-list prefix-list** *list-name* {**in**[*interface-type interface-number*] | **out** *routing-process* [*as-number*]} **Example:** `Device(config-router-af)# distribute-list prefix-list PL1 in Ethernet0/0`	Applies a prefix list to OSPFv3 routing updates that are received or sent on an interface.
Step 11	**maximum-paths** *number-paths* **Example:** `Device(config-router-af)# maximum-paths 4`	Controls the maximum number of equal-cost routes that a process for OSPFv3 routing can support.
Step 12	**summary-prefix** *prefix* [**not-advertise** | **tag** *tag-value*] **Example:** `Device(config-router-af)# summary-prefix FEC0::/24`	Configures an IPv6 summary prefix in OSPFv3.

Configuring Route Redistribution in OSPFv3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **address-family ipv6 unicast**
5. **redistribute** source-*protocol* [*process-id*] [*options*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router ospfv3** [*process-id*] **Example:** `Router(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.

Step 4	**address-family ipv6 unicast**
	Example:
	Example:
	`or`
	Example:
	address-family ipv4
	unicast
	Example:
	`Router(config-router)# address-family ipv6 unicast`
	Example:
	Example:
	`or`
	Example:
	`Router(config-router)# address-family ipv4 unicast`
		Enters IPv6 address family configuration mode for OSPFv3.
		or
		Enters IPv4 address family configuration mode for OSPFv3.

Step 5	**redistribute** source-*protocol* [*process-id*] [*options*] **Example:**	Redistributes IPv6 and IPv4 routes from one routing domain into another routing domain.

Enabling OSPFv3 on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ospfv3** *process-id* **area** *area-ID* {**ipv4** | **ipv6**} [**instance** *instance-id*]

- **ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface type and number, and places the device in interface configuration mode.

Step 4	Do one of the following:
	• **ospfv3** *process-id* **area** *area-ID* {**ipv4** | **ipv6**} [**instance** *instance-id*]
	• **ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*]
	Example:
	`Device(config-if)# ospfv3 1 area 1 ipv4`
	Example:
	`Device(config-if)# ipv6 ospf 1 area 0`
		Enables OSPFv3 on an interface with the IPv4 or IPv6 AF.
		or
		Enables OSPFv3 on an interface.

Defining an OSPFv3 Area Range for the IPv6 or IPv4 Address Family

The cost of the summarized routes will be the highest cost of the routes being summarized. For example, if the following routes are summarized:

```
OI  2001:DB8:0:7::/64 [110/20]
     via FE80::A8BB:CCFF:FE00:6F00, GigabitEthernet0/0/0
OI  2001:DB8:0:8::/64 [110/100]
     via FE80::A8BB:CCFF:FE00:6F00, GigabitEthernet0/0/0
OI  2001:DB8:0:9::/64 [110/20]
     via FE80::A8BB:CCFF:FE00:6F00, GigabitEthernet0/0/0
```

They become one summarized route, as follows:

```
OI  2001:DB8::/48 [110/100]
     via FE80::A8BB:CCFF:FE00:6F00, GigabitEthernet0/0/0
```

Before you begin

OSPFv3 routing must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **address-family ipv6 unicast**
5. **area** *area-ID* **range** *ipv6-prefix*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Router(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.

Step 4	**address-family ipv6 unicast**
	Example:
	Example:
	`or`
	Example:
	address-family ipv4
	unicast
	Example:
	`Router(config-router)# address-family ipv6 unicast`
	Example:
		Enters IPv6 address family configuration mode for OSPFv3.
		or
		Enters IPv4 address family configuration mode for OSPFv3.

	Command or Action	Purpose
	Example: or **Example:** `Router(config-router)# address-family ipv4 unicast`	
Step 5	**area** *area-ID* **range** *ipv6-prefix* **Example:** `Router(config-router-af)# area 1 range 2001:DB8:0:0::0/128`	Configures OSPFv3 area parameters.

Defining an OSPFv3 Area Range

This task can be performed in releases prior to Cisco IOS XE Release 3.4S.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **area** *area-id* **range** *ipv6-prefix* / *prefix-length* **advertise** | **not-advertise**] [**cost** *cost*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router ospf** *process-id* **Example:** `Router(config)# ipv6 router ospf 1`	Enables OSPFv3 router configuration mode.
Step 4	**area** *area-id* **range** *ipv6-prefix* / *prefix-length* **advertise**	**not-advertise**] [**cost** *cost*] **Example:**	Consolidates and summarizes routes at an area boundary.

	Command or Action	Purpose
	`Router(config-rtr)# area 1 range 2001:DB8::/48`	

Configuration Examples for OSPFv3 Address Families

Example: Configuring OSPFv3 Address Families

```
Device# show ospfv3
 Routing Process "ospfv3 1" with ID 10.0.0.1
 Supports IPv6 Address Family
 Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 Graceful restart helper support enabled
 Reference bandwidth unit is 100 mbps
 Relay willingness value is 128
 Pushback timer value is 2000 msecs
 Relay acknowledgement timer value is 1000 msecs
 LSA cache Disabled : current count 0, maximum 1000
 ACK cache Disabled : current count 0, maximum 1000
 Selective Peering is not enabled
 Hello requests and responses will be sent multicast
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Related Topic	Document Title
OSPFv3 Address Families	" *OSPF Forwarding Address Suppression in Translated Type-5 LSAs* " module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 Address Families

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 228: Feature Information for OSPFv3 Address Families

Feature Name	Releases	Feature Information
OSPFv3 Address Families	Cisco IOS XE Release 3.4S	

Feature Name	Releases	Feature Information
		The OSPFv3 address families feature enables IPv4 and IPv6 unicast traffic to be supported with a single network topology. The following commands were introduced or modified: **address-family ipv4 (OSPFv3)**, **address-family ipv6 (OSPFv3)**, **area (OSPFv3)**, **auto-cost (OSPFv3)**, **bfd all-interfaces (OSPFv3)**, **clear ospfv3 counters**, **clear ospfv3 force-spf**, **clear ospfv3 process**, **clear ospfv3 redistribution**, **clear ospfv3 traffic**, **debug ospfv3**, **debug ospfv3 database-timer rate-limit**, **debug ospfv3 events**, **debug ospfv3 lsdb**, **debug ospfv3 packet**, **debug ospfv3 spf statistic**, **default (OSPFv3)**, **default-information originate (OSPFv3)**, **default-metric (OSPFv3)**, **distance (OSPFv3)**, **distribute-list prefix-list (OSPFv3)**, **event-log (OSPFv3)**, **log-adjacency-changes (OSPFv3)**, **maximum-paths (OSPFv3)**, **ospfv3 area**, **ospfv3 authentication**, **ospfv3 bfd**, **ospfv3 cost**, **ospfv3 database-filter**, **ospfv3 dead-interval**, **ospfv3 demand-circuit**, **ospfv3 encryption**, **ospfv3 flood-reduction**, **ospfv3 hello-interval**, **ospfv3 mtu-ignore**, **ospfv3 network**, **ospfv3 priority**, **ospfv3 retransmit-interval**, **ospfv3 transmit-delay**, **passive-interface (OSPFv3)**, **queue-depth (OSPFv3)**, **redistribute (OSPFv3)**, **router ospfv3**, **router-id (OSPFv3)**, **show ospfv3 border-routers**, **show ospfv3 database**, **show ospfv3 events**, **show ospfv3 flood-list**, **show ospfv3 graceful-restart**, **show ospfv3 interface**, **show ospfv3 max-metric**, **show ospfv3 neighbor**, **show ospfv3 request-list**, **show ospfv3**

Feature Name	Releases	Feature Information
		retransmission-list, **show ospfv3 statistics**, **show ospfv3 summary-prefix**, **show ospfv3 timers rate-limit**, **show ospfv3 traffic**, **show ospfv3 virtual-links**, **summary-prefix (OSPFv3)**, **timers pacing flood (OSPFv3)**, **timers pacing lsa-group (OSPFv3)**, **timers pacing retransmission (OSPFv3)**.

Table 229: Feature Information for OSPFv3 Address Families

Feature Name	Releases	Feature Information
OSPFv3 Address Families	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 207

OSPFv3 Authentication Trailer

The OSPFv3 Authentication Trailer feature as specified in RFC 7166 provides a mechanism to authenticate Open Shortest Path First version 3 (OSPFv3) protocol packets as an alternative to existing OSPFv3 IPsec authentication.

Information About OSPFv3 Authentication Trailer

Overview of OSPFv3 Authentication Trailer

Prior to the OSPFv3 Authentication Trailer, OSPFv3 IPsec as defined in RFC 4552 was the only mechanism for authenticating protocol packets. The OSPFv3 Authentication Trailer feature defines an alternative mechanism to authenticate OSPFv3 protocol packets that additionally provides a packet replay protection via sequence number and does not have any platform dependencies.

To perform non-IPsec cryptographic authentication, OSPFv3 devices append a special data block, that is, Authentication Trailer, to the end of the OSPFv3 packets. The length of the Authentication Trailer is not included in the length of the OSPFv3 packet but is included in the IPv6 payload length. The Link-Local Signaling (LLS) block is established by the L-bit setting in the “OSPFv3 Options” field in OSPFv3 hello and database description packets. If present, the LLS data block is included along with the OSPFv3 packet in the cryptographic authentication computation.

A new Authentication Trailer (AT)-bit is introduced into the OSPFv3 Options field. OSPFv3 devices must set the AT-bit in OSPFv3 Hello and Database Description packets to indicate that all the packets on this link will include an Authentication Trailer. For OSPFv3 Hello and Database Description packets, the AT-bit indicates the AT is present. For other OSPFv3 packet types, the OSPFv3 AT-bit setting from the OSPFv3 Hello/Database Description setting is preserved in the OSPFv3 neighbor data structure. OSPFv3 packet types that do not include an OSPFv3 Options field will use the setting from the neighbor data structure to determine whether or not the AT is expected. The AT-bit must be set in all OSPFv3 Hello and Database Description packets that contain an Authentication Trailer.

To configure the Authentication Trailer, OSPFv3 utilizes existing Cisco IOS **key chain** command. For outgoing OSPFv3 packets, the following rules are used to select the key from the key chain:

- Select the key that is the last to expire.
- If two keys have the same stop time, select the one with the highest key ID.

The security association (SA) ID maps to the authentication algorithm and the secret key, which is used to generate and verify the message digest. The following authentication algorithms are supported:

- HMAC-SHA-1
- HMAC-SHA-256
- HMAC-SHA-384
- HMAC-SHA-512

If the authentication is configured but the last valid key is expired, then the packets are sent using the key. A syslog message is also generated. If no valid key is available then the packet is sent without the authentication trailer. When packets are received, the key ID is used to look up the data for that key. If the key ID is not found in the key chain or if the SA is not valid, the packet is dropped. Otherwise, the packet is verified using the algorithm and the key that is configured for the key ID. Key chains support rollover using key lifetimes. A new key can be added to a key chain with the send start time set in the future. This setting allows the new key to be configured on all devices before the keys are actually used.

The hello packets have higher priority than any other OSPFv3 packets and therefore can get re-ordered on the outgoing interface. This reordering can create problems with sequence number verification on neighboring devices. To prevent sequence mismatch, OSPFv3 verifies the sequence number separately for each packet type.

See RFC 7166 for more details on the authentication procedure.

Note If you receive packets that come in a non-decreasing sequence, the system displays an authentication error. This is not an error, and you can ignore this authentication error message. No other action is required from your end.

How to Configure OSPFv3 Authentication Trailer

Configuring OSPFv3 Authentication Trailer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ospfv3** [*pid*] [**ipv4** | **ipv6**] **authentication** {**key-chain** *chain-name* | **null**}
5. **router ospfv3** [*process-id*]
6. **address-family ipv6 unicast vrf** *vrf-name*
7. **area** *area-id* **authentication** {**key-chain** *chain-name* | **null**}
8. **area** *area-id* **virtual-link** *router-id* **authentication key-chain** *chain-name*

9. **area** *area-id* **sham-link** *source-address destination-address* **authentication key-chain** *chain-name*
10. **authentication mode** {**deployment** | **normal**}
11. **end**
12. **show ospfv3 interface**
13. **show ospfv3 neighbor** [*detail*]
14. **debug ospfv3 vrf authentication**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 2/0`	Specifies the interface type and number.
Step 4	**ospfv3** [*pid*] [**ipv4** | **ipv6**] **authentication** {**key-chain** *chain-name* | **null**} **Example:** `Device(config-if)# ospfv3 1 ipv4 authentication key-chain ospf-1`	Specifies the authentication type for an OSPFv3 instance.
Step 5	**router ospfv3** [*process-id*] **Example:** `Device(config-if)# router ospfv3 1`	Enters OSPFv3 router configuration mode.
Step 6	**address-family ipv6 unicast vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv6 unicast vrf vrf1`	Configures the IPv6 address family in the OSPFv3 process and enters IPv6 address family configuration mode.
Step 7	**area** *area-id* **authentication** {**key-chain** *chain-name* | **null**} **Example:** `Device(config-router-af)# area 1 authentication key-chain ospf-chain-1`	Configures the authentication trailer on all interfaces in the OSPFv3 area.
Step 8	**area** *area-id* **virtual-link** *router-id* **authentication key-chain** *chain-name* **Example:**	Configures the authentication for virtual links.

	Command or Action	Purpose
	`Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-chain-1`	
Step 9	**area** *area-id* **sham-link** *source-address destination-address* **authentication key-chain** *chain-name* **Example:** `Device(config-router-af)# area 1 sham-link 1.1.1.1 1.1.1.0 authentication key-chain ospf-chain-1`	Configures the authentication for sham links.
Step 10	**authentication mode** {**deployment** | **normal**} **Example:** `Device(config-router-af)# authentication mode deployment`	Specifies the type of authentication used for the OSPFv3 instance. The **deployment** keyword provides adjacency between configured and unconfigured authentication devices. In deployment mode, a router processes packets as following: • The ospf checksum is calculated for the outgoing packets even if the authentication trailer is configured. • However, for the incoming packets the packets without authentication trailer or the wrong authentication hash packets get dropped. In this mode, the show ospfv3 neighbor detail command shows the last packet authentication status which can be used to verify the authentication trailer method.
Step 11	**end** **Example:** `Device(config-router-af)# end`	Exits IPv6 address family configuration mode and returns to privileged EXEC mode.
Step 12	**show ospfv3 interface** **Example:** `Device# show ospfv3`	(Optional) Displays OSPFv3-related interface information.
Step 13	**show ospfv3 neighbor** [*detail*] **Example:** `Device# show ospfv3 neighbor detail`	(Optional) Displays OSPFv3 neighbor information on a per-interface basis.
Step 14	**debug ospfv3 vrf authentication** **Example:** `Device# debug ospfv3 vrf authentication`	(Optional) Displays debugging information for OSPFv3.

Configuration Examples for OSPFv3 Authentication Trailer

Example: Configuring OSPFv3 Authentication Trailer

```
interface GigabitEthernet 0/0
 ospfv3 1 ipv4 authentication key-chain ospf-1
 router ospfv3 1
  address-family ipv6 unicast vrf vrf1
   area 1 authentication key-chain ospf-1
   area 1 virtual-link 1.1.1.1 authentication key-chain ospf-1
   area 1 sham-link 1.1.1.1 authentication key-chain ospf-1
   authentication mode deployment
!
key chain ospf-1
key 1
   key-string ospf
   cryptographic-algorithm hmac-sha-512
!
```

Example: Verifying OSPFv3 Authentication Trailer

The following examples show the output of the **show ospfv3** commands.

```
Device# show ospfv3
 OSPFv3 1 address-family ipv6
 Router ID 1.1.1.1
…
RFC1583 compatibility enabled
 Authentication configured with deployment key lifetime
 Active Key-chains:
  Key chain mama: Send key 1, Algorithm HMAC-SHA-256, Number of interfaces 1
    Area BACKBONE(0)


Device# show ospfv3 neighbor detail

OSPFv3 1 address-family ipv6 (router-id 2.2.2.2)

 Neighbor 1.1.1.1
    In the area 0 via interface GigabitEthernet0/0
    Neighbor: interface-id 2, link-local address FE80::A8BB:CCFF:FE01:2D00
    Neighbor priority is 1, State is FULL, 6 state changes
    DR is 2.2.2.2 BDR is 1.1.1.1
    Options is 0x000413 in Hello (V6-Bit, E-Bit, R-Bit, AT-Bit)
    Options is 0x000413 in DBD (V6-Bit, E-Bit, R-Bit, AT-Bit)
    Dead timer due in 00:00:33
    Neighbor is up for 00:05:07
    Last packet authentication succeed
    Index 1/1/1, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

```
Device# show ospfv3 interface

GigabitEthernet0/0 is up, line protocol is up
…
  Cryptographic authentication enabled
    Sending SA: Key 25, Algorithm HMAC-SHA-256 - key chain ospf-keys
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

Additional References for OSPFv3 Authentication Trailer

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Configuring OSPF features	IP Routing: OSPF Configuration Guide

Standards and RFCs

Related Topic	Document Title
RFC for Supporting Authentication Trailer for OSPFv3	RFC 6506
RFC for Authentication/Confidentiality for OSPFv3	RFC 4552

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for OSPFv3 Authentication Trailer

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 230: Feature Information for OSPFv3 Authentication Trailer

Feature Name	Releases	Feature Information
OSPFv3 Authentication Trailer	Cisco IOS XE Release 3.11S	The OSPFv3 Authentication Trailer feature as specified in RFC 6506 provides a mechanism to authenticate OSPFv3 protocol packets as an alternative to existing OSPFv3 IPsec authentication. The following commands were introduced or modified: **ospfv3 authentication key-chain**, **authentication mode**, **debug ospfv3 vrf authentication**.

Table 231: Feature Information for OSPFv3 Authentication Trailer

Feature Name	Releases	Feature Information
OSPFv3 Authentication Trailer	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 208

Autoroute Announce and Forwarding Adjacencies For OSPFv3

The Autoroute Announce and Forwarding Adjacencies for OSPFv3 feature advertises IPv6 routes over MPLS/TE IPv4 tunnels. This module describes how to configure the Autoroute Announce and Forwarding Adjacencies for OSPFv3 feature.

Prerequisites for Autoroute Announce and Forwarding Adjacencies For OSPFv3

- OSPFv3 must be configured in your network.
- Cisco Express Forwarding (CEF) must be enabled.
- MPLS/TE tunnels must be configured.

Restrictions for Autoroute Announce and Forwarding Adjacencies For OSPFv3

- Autoroute announce and forwarding adjacency cannot be configured together in a same interface.
- When an autoroute announce is used, OSPFv3 does not advertise the tunnel.
- When forwarding adjacencies are used, OSPFv3 advertises the tunnel link in an LSA.

Information About Autoroute Announce and Forwarding Adjacencies For OSPFv3

Overview of Autoroute Announce and Forwarding Adjacencies For OSPFv3

The OSPFv3 support for Forwarding Adjacencies over MPLS Traffic Engineered Tunnels feature adds OSPFv3 support to the Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) tunnels feature, which allows a network administrator to handle a traffic engineering, MPLS tunnel as a link in an Interior Gateway Protocol (IGP) network based on the shortest path first (SPF) algorithm. An OSPFv3 forwarding adjacency can be created between routers in the same area.

OSPFv3 includes MPLS TE tunnels in the OSPFv3 router link-state advertisement (LSA) in the same way that other links appear for purposes of routing and forwarding traffic. The user can assign an OSPFv3 cost to the tunnel to give it precedence over other links. Other networking devices will see the tunnel as a link in addition to the physical link.

OSPFv3 uses Autoroute Announce (AA) or Forwarding Adjacencies (FA) feature to install IPv6 routes over MPLS/TE IPv4 tunnels into the IPv6 routing table . The TE tunnels are created using IPv4, and requires the use of a routing protocol other than OSPFv3. OSPFv2 is used as the IPv4 IGP and provides data which TE uses to create the tunnels.

OSPFv3 is configured on the TE tunnel interfaces for either autoroute-annouce or forwarding-adjacency. It is also must be configured in router mode to advertise the address of the loopback interface which TE is using for the tunnels that terminate on the router. That address is advertised in the TE LSA .

How to Configure Autoroute Announce and Forwarding Adjacencies For OSPFv3

Configuring Autoroute Announce and Forwarding Adjacencies For OSPFv3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **interface** *type number*
5. **ip address** *ip-address-mask*
6. **no shutdown**
7. **exit**
8. **interface** *type number*
9. **ospfv3** *pid af* **mpls traffic-eng autoroute announce area** *aid*
10. **ospfv3** *pid af* **mpls traffic-eng autoroute metric** {*metric* | **absolute** *metric* | **relative** *delta*}
11. **ip ospf cost** *cost*

12. **exit**
13. **interface** *type number*
14. **ospfv3** *pid af* **mpls traffic-eng forwarding-adj area***aid*
15. **ospfv3**[*pid* [*af*]] **mpls traffic-eng forwarding-adj** *interface ID* [**local** *ID*] [**nbr** *ID*]
16. **ip ospf cost** *cost*
17. **exit**
18. **router ospfv3** *router-ID*
19. **address-family ipv4 unicast** [**vrf** *vrf-name*]
20. **area** *aid* **mpls traffic-engineering tunnel-tail** *af interface type*
21. **exit**
22. **show ospfv3 database**
23. **show ospfv3 mpls traffic-eng**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:** `Device(config)# ip cef distributed`	Enables distributed Cisco Express Forwarding operation.
Step 4	**interface** *type number* **Example:** `Device(config)# interface tunnel 0`	Configures an interface type and enters interface configuration mode.
Step 5	**ip address** *ip-address-mask* **Example:** `Device (config-if)# ip address 192.108.1.27 255.255.255.0`	Sets a primary or secondary IP address for the specified interface.
Step 6	**no shutdown** **Example:** `Device (config-if)# no shutdown`	Disables all functions on the specified interface.
Step 7	**exit** **Example:** `Device (config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Device (config)# interface loopback 0`	Enables loopback interface and enters interface configuration mode.
Step 9	**ospfv3** *pid af* **mpls traffic-eng autoroute announce area** *aid* **Example:** `Device(config-if)# ospfv3 1 af mpls traffic-eng autoroute announce area 1`	Enable Open Shortest Path First version 3 (OSPFv3) on an interface with the IP address family (AF).
Step 10	**ospfv3** *pid af* **mpls traffic-eng autoroute metric** {*metric* | **absolute** *metric* | **relative** *delta*} **Example:** `Device(config-if)# ospfv3 1 af mpls traffic-eng autoroute metric 1`	Specifies the MPLS traffic engineering auto route metric value for the SPF calculation.
Step 11	**ip ospf cost** *cost* **Example:** `Device(config-if)# ip ospf cost 60`	Explicitly specifies the cost of sending a packet on an OSPF interface.
Step 12	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 13	**interface** *type number* **Example:** `Device (config)# interface tunnel 1`	Enables tunnel interface and enters interface configuration mode.
Step 14	**ospfv3** *pid af* **mpls traffic-eng forwarding-adj area***aid* **Example:** `Device(config-if)# ospfv3 1 af mpls traffic-eng forwarding-adj area 1`	Configure an MPLS traffic engineering forwarding adjacency.
Step 15	**ospfv3**[*pid* [*af*]] **mpls traffic-eng forwarding-adj** *interface ID* [**local** *ID*] [**nbr** *ID*] **Example:** `Device(config-if)# ospfv3 1 af mpls traffic-eng forwarding-adj 1`	Specifies the MPLS traffic engineering forwarding adjacency for the SPF calculation.
Step 16	**ip ospf cost** *cost* **Example:** `Device(config-if)# ip ospf cost 55`	Explicitly specifies the cost of sending a packet on an OSPF interface.

	Command or Action	Purpose
Step 17	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 18	**router ospfv3** *router-ID* **Example:** `Device(config)# router ospfv3 18`	Enters OSPFv3 router configuration mode.
Step 19	**address-family ipv4 unicast** [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Configures the IPv4 address family in the OSPFv3 process and enters IPv4 address family configuration mode.
Step 20	**area** *aid* **mpls traffic-engineering tunnel-tail** *af interface type* **Example:** `Device(config-router-af)# area 1 mpls traffic-engineering tunnel-tail af loopback`	Configures OSPFv3 on the tail end of the traffic engineering tunnels.
Step 21	**exit** **Example:** `Device(config-router-af)# exit`	Exits address family configuration mode and returns to global configuration mode.
Step 22	**show ospfv3 database** **Example:** `Device(config)# show ospfv3 database`	(Optional) Displays list of information related to the OSPFv3 database for a specific router.
Step 23	**show ospfv3 mpls traffic-eng** **Example:** `Device(config)# show ospfv3 mpls traffic-eng`	(Optional) Displays autoroute announce, forwarding adjacency, and tunnel-tail information related to OSPFv3.

Configuration Examples for Autoroute Announce and Forwarding Adjacencies For OSPFv3

Example: Configuring Autoroute Announce and Forwarding Adjacencies For OSPFv3

```
!
ip cef distributed
interface tunnel 0
 ip address 192.108.1.27 255.255.255.0
```

```
 no shutdown

interface loopback 0
 ospfv3 1 af mpls traffic-eng autoroute announce area 1
 ospfv3 1 af mpls traffic-eng autoroute metric 1
 ip ospf cost 60

interface tunnel 1
 ospfv3 1 af mpls traffic-eng forwarding-adj area 1
 ospfv3 1 af mpls traffic-eng forwarding-adj nbr 1
 ip ospf cost 55

router ospfv3 18
 address-family ipv4 unicast
   area 1 mpls traffic-engineering tunnel-tail af loopback

   !
  !
!
```

Additional References for Autoroute Announce and Forwarding Adjacencies For OSPFv3

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Configuring OSPF features	IP Routing: OSPF Configuration Guide

Standards and RFCs

Related Topic	Document Title
Advertising a Router's Local Addresses in OSPF Traffic Engineering (TE) Extensions	RFC5786
Traffic Engineering Extensions to OSPF Version 3	RFC5329
Traffic Engineering (TE) Extensions to OSPF Version 2	RFC3630

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Autoroute Announce and Forwarding Adjacencies For OSPFv3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 232: Feature Information for Autoroute Announce and Forwarding Adjacencies For OSPFv3

Feature Name	Releases	Feature Information
Autoroute Announce and Forwarding Adjacencies For OSPFv3	Cisco IOS XE Release 3.12S	The Autoroute Announce and Forwarding Adjacencies For OSPFv3 feature advertises IPv6 routes over MPLS/TE IPv4 tunnels. The following commands were introduced or modified: **ospfv3 af mpls traffic-eng autoroute announce area** , **ospfv3 mpls traffic-eng autoroute metric**, **ospfv3 mpls traffic-eng forwarding-adj area** .

Table 233: Feature Information for Autoroute Announce and Forwarding Adjacencies For OSPFv3

Feature Name	Releases	Feature Information
Autoroute Announce and Forwarding Adjacencies For OSPFv3	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 209

OSPFv3 Autoroute Exclude

OSPFv3 Autoroute Exclude feature allows you to use specific destinations and prefix-list to specify a list of prefixes that are routed using native paths instead of TE tunnels for packet transport. The rest of the prefixes can still be set to use TE tunnels. Prefixes that are excluded do not use a TE tunnel path. IPv6 routes over TE tunnels are supported by OSPFv3 using Autoroute Announce (AA) or Forwarding Adjacencies (FA).

This module describes how to configure the OSPFv3 Autoroute Exclude feature.

Prerequisites for OSPFv3 Autoroute Exclude

- Open Shortest Path First (OSPF) must be configured in your network.
- Cisco Express Forwarding (CEF) must be enabled.
- Multiprotocol Label Switching (MPLS) TE tunnels must be configured.
- Auto route announce and forwarding adjacencies must be configured. You can configure either auto route announce or forwarding adjacencies on an interface. You cannot configure them both on the same interface.

Information About OSPFv3 Autoroute Exclude

Overview of OSPFv3 Autoroute Exclude

The auto route feature is an IP routing method that forces OSPF to use MPLS TE tunnels to build paths for IP traffic routes. The auto route feature enables all routes to use TE Tunnels, even if there is an alternate non-TE path available for that route.

The OSPFv3 Autoroute Exclude feature allows specific IPv6 destinations or prefixes to avoid TE tunnels, while other prefixes can still be configured to use TE tunnels. Prefixes that are excluded do not use a TE

tunnel path. Only native non-TE paths are downloaded to RIB for such routes. IPv6 routes over TE tunnels are supported by OSPFv3 using auto route announce (AA) or forwarding adjacencies (FA).

The auto route exclude option is configured under the router OSPF configuration mode by using a prefix list. IP addresses and prefixes that are members of this prefix list are excluded from TE tunnels, even when the auto route is enabled on them. If the IP addresses or prefixes are added to the prefix list, they are dynamically routed without passing through the TE tunnel. If the IP addresses or prefixes are removed from the prefix list, they are dynamically rerouted back on the TE tunnel path.

See the *Autoroute Announce and Forwarding Adjacencies For OSPFv3* module in *IP Routing: OSPF Configuration Guide* for details on configuring auto route announce and forwarding adjacencies For OSPFv3.

How to Configure OSPFv3 Autoroute Exclude

Configuring OSPFv3 Autoroute Exclude

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** *process-ID*
4. **address-family ipv6 unicast**
5. **mpls traffic-engineering autoroute-exclude prefix-list** *prefix-list-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** *process-ID* **Example:** `Device(config)# router ospfv3 18`	Configures OSPFv3 routing process and enters OSPF router configuration mode.
Step 4	**address-family ipv6 unicast** **Example:** `Device(config-router)# address-family ipv6 unicast`	Enters IPv6 address family configuration mode for OSPFv3.
Step 5	**mpls traffic-engineering autoroute-exclude prefix-list** *prefix-list-name*	Allows specific destinations and prefixes to avoid routing through TE tunnels.

	Command or Action	Purpose
	Example: `Device(config-router-af)# mpls traffic-engineering autoroute-exclude prefix-list kmd`	• Prefixes that are excluded do not use a TE tunnel path.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for OSPFv3 Autoroute Exclude

Example: Configuring OSPFv3 Autoroute Exclude

```
!
router ospfv3 18
 address-family ipv6 unicast
  mpls traffic-engineering autoroute-exclude prefix-list kmd

!
```

Additional References for OSPFv3 Autoroute Exclude

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Configuring OSPF	*IP Routing: OSPF Configuration Guide*
Autoroute Announce and Forwarding Adjacencies For OSPFv3	*IP Routing: OSPF Configuration Guide*
Configuring Basic Cisco Express Forwarding	*IP Switching: Cisco Express Forwarding Configuration Guide*
MPLS Traffic Engineering Tunnel Source	*MPLS Traffic Engineering Path Calculation and Setup Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for OSPFv3 Autoroute Exclude

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 210

OSPFv2 IP FRR Local Microloop Avoidance

The OSPFv2 IP FRR Local Microloop Avoidance feature helps to avoid local microloop that happens between a node and its neighbor where the link-down event occurred. This document explains how to configure the OSPFv2 IP FRR Local Microloop Avoidance feature.

Information About OSPFv2 IP FRR Local Microloop Avoidance

Overview of OSPFv2 IP FRR Local Microloop Avoidance

IP fast reroute (IPFRR) provides rapid convergence during the link-down events by moving the traffic to a pre computed backup path until the regular convergence mechanisms move the traffic to the newly found best path referred to as the post-convergence path.

Once the traffic is moved to the post-convergence path, it is inclined to a microloop. Microloops are formed as a result of the fact that each node on the path does its calculation at different times and independently of other nodes. If certain nodes converge and sends traffic to a neighbor node, which has not converged yet, traffic may be looped between these two nodes.

Microloops are formed between the router where the failure is detected and its neighbors. Local microloops are created in cases where there is no local loop-free alternate (LFA) backup available in ring or square topologies. In such topologies, remote LFA provides a backup, but the fast-convergence benefit of the remote LFA cannot be completely utilized due to the high probability of the local microloop creation. Avoiding the local micro loop provides a significant improvement in the fast convergence in the ring and square topologies.

Note Microloop avoidance is automatically enabled as soon as remote LFA (rLFA) is enabled.

When using microloop avoidance for prefixes (for which a repair path has been installed in the forwarding plane), the OSPFv2 IP FRR Local Microloop Avoidance feature is enabled when the forwarding plane is

triggered to switch to using a pre installed repair path. The local microloop avoidance for the link-down event supports the following triggers:

- Interface down event.
- Adjacency down event due to the Bidirectional Forwarding Detection (BFD) session down.

If microloop avoidance is used regardless of whether a repair path has been installed in the forwarding plane, then in addition the third trigger is used:

- Adjacency down event due to neighbor hold time expiration.

When the neighbor reports loss of adjacency to the local system in its link state neighbor advertisements, the value of using microloop avoidance depends on whether the remote event that caused loss of adjacency on the neighbor is detectable by the local forwarding plane (that is, whether the forwarding plane will react and switch to using pre programmed repair paths).

How to Configure OSPFv2 IP FRR Local Microloop Avoidance

Configuring OSPFv2 IP FRR Local Microloop Avoidance

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **microloop avoidance** [**protected** | **disable**]
5. **microloop avoidance rib-update-delay** *delay-period*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Configures an OSPF routing process and enters router configuration mode.
Step 4	**microloop avoidance** [**protected**	**disable**] **Example:**	Configures the local microloop avoidance between a node and its neighbor where the link-down event has occurred.

	Command or Action	Purpose
	`Device(config-router)# microloop avoidance protected`	• When the **protected** keyword is used, the local microloop avoidance is only applied to prefixes that have a valid backup path. • When the **disable** keyword is used, the local microloop avoidance is disabled if it is enabled automatically earlier.
Step 5	**microloop avoidance rib-update-delay** *delay-period* **Example:** `Device(config-router)# microloop avoidance rib-update-delay 6500`	Delays the local microloop avoidance as per the configured delay period.
Step 6	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to privileged EXEC mode.

Configuration Examples for OSPFv2 IP FRR Local Microloop Avoidance

Example: Configuring OSPFv2 IP FRR Local Microloop Avoidance

```
router ospf 10
 microloop avoidance protected
microloop avoidance rib-update-delay 6500
!
```

Additional References for OSPFv2 IP FRR Local Microloop Avoidance

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Configuring OSPF features	IP Routing: OSPF Configuration Guide

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for OSPFv2 IP FRR Local Microloop Avoidance

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 234: Feature Information for OSPFv2 IP FRR Local Microloop Avoidance

Feature Name	Releases	Feature Information
OSPFv2 IP FRR Local Microloop Avoidance	Cisco IOS XE Release 3.11S 15.4(1)S	The OSPFv2 IP FRR Local Microloop Avoidance feature helps to avoid local microloop that happens between a node and its neighbor where the link-down event occurred. The following commands were introduced or modified: **microloop avoidance**, **microloop avoidance rib-update-delay**.

Table 235: Feature Information for OSPFv2 IP FRR Local Microloop Avoidance

Feature Name	Releases	Feature Information
OSPFv2 IP FRR Local Microloop Avoidance	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 211

OSPFv2-OSPF Live-Live

The OSPFv2-OSPF Live-Live feature delivers multicast streams over non overlapping paths to various applications. The multicast traffic is split into multiple streams at the beginning of a protected network. All streams flow over non overlapping paths so that when a link failure occurs on one path, multicast traffic is still delivered through other paths. All streams are merged back at the end of the protected network. This module describes how to configure the OSPFv2-OSPF Live-Live feature.

Information About OSPFv2-OSPF Live-Live

Overview of OSPFv2-OSPF Live-Live

Many new applications driving the growth of networking market are multicast based. Applications such as Internet Protocol television (IPTV) are typically associated with simultaneously delivering massive amount of sensitive data streams to large audiences. Packet drop is a critical issue in multimedia traffic. There is a demand to reduce multicast traffic loss to the range of milliseconds or to zero packet loss. The zero packet loss solution for multicast in case of single link failure is also known as live-live.

In a live-live network, multicast streams (typically two flows) form their own reverse path forwarding (RPF)/shortest path trees (SPT) over diversified physical links, so that failure on one link does not affect multicast traffic on other link. The existing multi topology technology in Cisco IOS software supports the multiple multicast topologies.

The OSPFv2-OSPF Live-Live feature enables the protocol independent multicast (PIM) to handle multiple multicast topologies. When a multicast topology is created and enabled on OSPF, IP prefixes on each topology are injected into topology-based Routing Information Base (RIB). PIM then decides which RIB to use for RPF lookup.

PIM RPF topology is a collection of routes used by PIM to perform the RPF operation when building shared or source trees. In a multi topology environment, multiple RPF topologies can be created in the same network. A particular source may be reachable in only one of the topologies or in several of them through different paths.

To select the RPF topology for a particular multicast distribution tree, consider the following:

1. Configure a policy that maps a group range to a topology. When RPF information needs to be resolved for the RP or the sources for a group within the range, the RPF lookup takes place in the specified topology. This can be used for PIM Sparse Mode (PIM-SM)/source-specific multicast (SSM)/Bidirectional(Bidir) PIM.
2. Configure a policy that maps a source prefix range to a topology. This can be used for PIM-SM and PIM-SSM.
3. Use the topology identified by the Join Attribute encoding in the received PIM packets.

The PIM Join Attribute extends PIM signaling to identify a topology that should be used when constructing a particular multicast distribution tree. For more details on the PIM Join Attribute, see PIM Multi-Topology ID (MT-ID) Join-Attribute IEEE draft.

How to Configure OSPFv2-OSPF Live-Live

Configuring OSPFv2-OSPF Live-Live

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**
4. **ip multicast rpf multitopology**
5. **global-address-family ipv4 multicast**
6. **topology** {*topology-A* | *topology-B*}
7. **exit**
8. **interface** *type number*
9. **ip address** *address mask*
10. **ip pim sparse-dense-mode**
11. **ip ospf** *process-id* **area** *area-id*
12. **topology ipv4 multicast** *topology-name*
13. **exit**
14. **router ospf** *process-id*
15. **network** *ip-adddress mask* **area** *area-id*
16. **address-family ipv4 multicast**
17. **topology** *topology-name* **tid** *topology-id*
18. **end**
19. **configure terminal**
20. **ip multicast topology multicast** *topology-name* **tid** *topology-id*
21. **ip multicast rpf select topology multicast** *topology-name access-list number*
22. **ip access-list extended** *access-list-number*
23. **permit ip any** *ip-adddress*
24. **end**
25. **show ip multicast topology multicast** *topology-name*

26. debug ip multicast topology

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** **Example:** `Device(config)# ip multicast-routing`	Enables IP multicast routing.
Step 4	**ip multicast rpf multitopology** **Example:** `Device(config)# ip multicast rpf multitopology`	Enables Multi Topology Routing (MTR) support for IP multicast routing.
Step 5	**global-address-family ipv4 multicast** **Example:** `Device(config)# global-address-family ipv4 multicast`	Enters global address family configuration mode and configures multi topology routing.
Step 6	**topology** {*topology-A* | *topology-B*} **Example:** `Device(config-af)# topology live-A`	Configures an OSPF process to route IP traffic under the specified topology instance.
Step 7	**exit** **Example:** `Device(config-af)# exit`	Exits address family configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0`	Configures an interface type and enters interface configuration mode.
Step 9	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.108.1.27 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 10	**ip pim sparse-dense-mode** **Example:** `Device(config-if)# ip pim sparse-dense-mode`	Enables PIM on an interface and treats the interface in either sparse mode or dense mode of operation, depending on which mode the multicast group operates in.

	Command or Action	Purpose
Step 11	**ip ospf** *process-id* **area** *area-id* **Example:** `Device(config-if)# ip ospf 10 area 0`	Enables OSPFv2 on an interface.
Step 12	**topology ipv4 multicast** *topology-name* **Example:** `Device(config-if)# topology ipv4 multicast live-A`	Configures a multi topology instance on an interface.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode. • Repeat Steps 9 to 12 to configure the next topology (**topology ipv4 multicast live-B**).
Step 14	**router ospf** *process-id* **Example:** `Device(config)# router ospf 102`	Enables OSPF routing and enters router configuration mode.
Step 15	**network** *ip-adddress mask* **area** *area-id* **Example:** `Device(config-router)# network 192.168.129.16 0.0.0.3 area 20`	Defines an interface on which OSPF runs and defines the area ID for that interface.
Step 16	**address-family ipv4 multicast** **Example:** `Device(config-router)# address-family ipv4 multicast`	Enters router address family configuration mode and configures OSPF to exchange IPv4 multicast prefixes.
Step 17	**topology** *topology-name* **tid** *topology-id* **Example:** `Device(config-router-af)# topology live-A tid 100`	Configures an OSPF process to route IP traffic under the specified topology instance. • Repeat this step to configure the OSPF process to route IP traffic under another topology instance (**topology live-B tid 200**).
Step 18	**end** **Example:** `Device(config-router-af)# end`	Exits router address family configuration mode and returns to privileged EXEC mode.
Step 19	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 20	**ip multicast topology multicast** *topology-name* **tid** *topology-id* **Example:** `Device(config)# ip multicast topology multicast live-A tid 100`	Configures topology selection for the multicast streams. • Repeat this step to configure another topology (**ip multicast topology multicast live-B tid 200**).

	Command or Action	Purpose
Step 21	**ip multicast rpf select topology multicast** *topology-name access-list number* **Example:** `Device(config)# ip multicast rpf select topology multicast topology live-A 111`	Associates a multicast topology with a multicast group with a specific route entry. • Repeat this step to associate the topology with another multicast group (**ip multicast rpf select topology multicast live-B 122**).
Step 22	**ip access-list extended** *access-list-number* **Example:** `Device(config)# ip access-list extended 111`	Defines an IP access list to enable filtering for packets with IP helper-address destinations and enters extended named access list configuration mode.
Step 23	**permit ip any** *ip-adddress* **Example:** `Device(config-ext-nacl)# permit ip any 203.0.113.1`	Sets condition to allow a packet to pass a named IP access list. • Repeat Steps 22 and 23 to define another IP access list and to set conditions to allow a packet to pass another named IP access list.
Step 24	**end** **Example:** `Device(config-ext-nacl)# end`	Exits extended named access list configuration mode and enters privileged EXEC mode.
Step 25	**show ip multicast topology multicast** *topology-name* **Example:** `Device# show ip multicast topology multicast live-A`	Displays topology information for multicast streams.
Step 26	**debug ip multicast topology** **Example:** `Device# debug ip multicast topology`	Enables debugging output for multicast stream topology.

Configuration Examples for OSPFv2-OSPF Live-Live

Example: Configuring OSPFv2-OSPF Live-Live

```
ip multicast-routing
!
ip multicast rpf multitopology

!
global-address-family ipv4 multicast
  topology live-A
  topology live-B

int gigabitethernet 1/0
 ip address 192.0.2.1 255.255.255.0
 ip pim sparse-dense-mode
 ip ospf 10 area 20
```

```
 topology ipv4 multicast live-A
!
int gigabitethernet 2/0
 ip address 192.0.2.2 255.255.255.0
 ip pim sparse-dense-mode
 ip ospf 11 area 21
 topology ipv4 multicast live-B
!
router ospf 1
 network 192.168.129.16 0.0.0.3 area 20
  address-family ipv4 multicast
  !!
   topology live-A tid 10
   topology live-B tid 20
!
!!
ip multicast topology multicast live-A tid 100
ip multicast topology multicast live-B tid 200
!
!!
ip multicast rpf select topology multicast live-A 111
ip multicast rpf select topology multicast live-B 122

!
 ip access-list extended 111
 permit ip any 203.0.113.254

ip access-list extended 122
 permit ip any 203.0.113.251
```

Additional References for OSPFv2-OSPF Live-Live

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Configuring OSPF features	IP Routing: OSPF Configuration Guide

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for OSPFv2-OSPF Live-Live

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 236: Feature Information for OSPFv2-OSPF Live-Live

Feature Name	Releases	Feature Information
OSPFv2-OSPF Live-Live	Cisco IOS XE Release 3.11S	The OSPFv2-OSPF Live-Live feature delivers multicast streams over non overlapping paths to various applications. The multicast traffic is split into multiple streams at the beginning of a protected network. All streams flow over non overlapping paths so that when a link failure occurs on one path, multicast traffic is still delivered through other paths. All streams are merged back at the end of the protected network. No commands were introduced or modified.

Table 237: Feature Information for OSPFv2-OSPF Live-Live

Feature Name	Releases	Feature Information
OSPFv2-OSPF Live-Live	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 212

OSPF Forwarding Address Suppression in Translated Type-5 LSAs

The OSPF Forwarding Address Suppression in Translated Type-5 LSAs feature causes a not-so-stubby area (NSSA) area border router (ABR) to translate Type-7 link state advertisements (LSAs) to Type-5 LSAs, but to use the address 0.0.0.0 for the forwarding address instead of that specified in the Type-7 LSA. This feature causes devices that are configured not to advertise forwarding addresses into the backbone to direct forwarded traffic to the translating NSSA ABRs.

Prerequisites for OSPF Forwarding Address Suppression

This document presumes that you have OSPF configured on the networking device; it does not document other steps to configure OSPF.

Information About OSPF Forwarding Address Suppression

Benefits of OSPF Forwarding Address Suppression

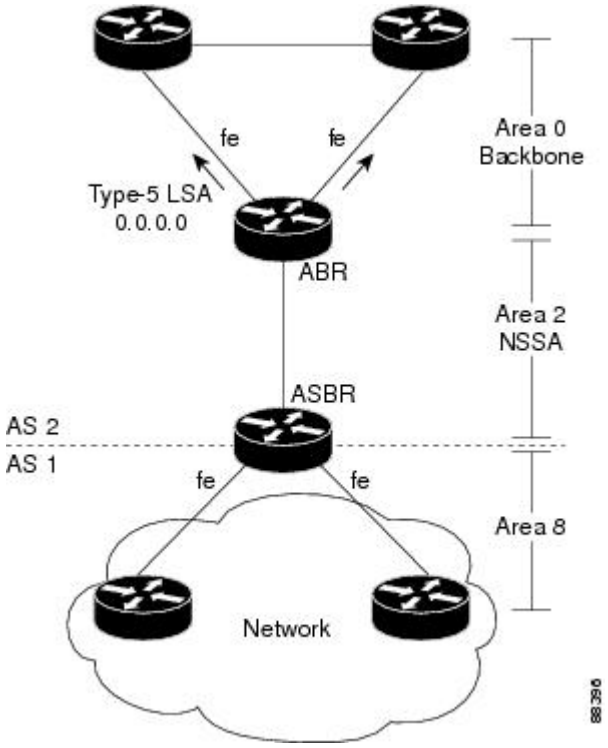
The OSPF Forwarding Address Suppression in Translated Type-5 LSAs feature causes an NSSA ABR to translate Type-7 LSAs to Type-5 LSAs, but use the 0.0.0.0 as the forwarding address instead of that specified in the Type-7 LSA. This feature causes devices that are configured not to advertise forwarding addresses into the backbone to direct forwarded traffic to the translating NSSA ASBRs.

When to Suppress OSPF Forwarding Address in Translated Type-5 LSAs

In the figure below, it would be advantageous to filter Area 2 addresses from Area 0 to minimize the number of routes introduced into the backbone (Area 0). However, using the **area range**command to consolidate and

summarize routes at the area boundary--filtering the Area 2 addresses--will not work because the Area 2 addresses include forwarding addresses for Type-7 LSAs that are generated by the ASBR. If these Type-7 LSA forwarding addresses have been filtered out of Area 0, the backbone routers cannot reach the prefixes advertised in the translated Type-5 LSAs (autonomous system external LSAs).

Figure 209: OSPF Forwarding Address Suppression in Translated Type-5 LSAs



This problem is solved by suppressing the forwarding address on the ABR so that the forwarding address is set to 0.0.0.0 in the Type-5 LSAs that were translated from Type-7 LSAs. A forwarding address set to 0.0.0.0 indicates that packets for the external destination should be forwarded to the advertising OSPF device, in this case, the translating NSSA ABR.

Before configuring this feature, consider the following caution.

Caution Configuring this feature causes the device to be noncompliant with RFC 1587. Also, suboptimal routing might result because there might be better paths to reach the destination's forwarding address. This feature should not be configured without careful consideration and not until the network topology is understood.

How to Suppress the OSPF Forwarding Address

Suppressing the OSPF Forwarding Address in Translated Type-5 LSAs

This task describes how to suppress the OSPF forwarding address in translated Type-5 LSAs. Before configuring this feature, consider the following caution.

Caution Configuring this feature causes the device to be noncompliant with RFC 1587. Also, suboptimal routing might result because there might be better paths to reach the destination's forwarding address. This feature should not be configured without careful consideration and not until the network topology is understood.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **area** *area-id* **nssa translate type7 suppress-fa**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 1`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process.
Step 4	**area** *area-id* **nssa translate type7 suppress-fa** **Example:** `Device(config-router)# area 10 nssa translate type7 suppress-fa`	Configures an area as a not-so-stubby-area (NSSA) and suppresses the forwarding address in translated Type-7 LSAs.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits configuration mode and returns to privileged EXEC mode.

Configuration Examples for OSPF Forwarding Address Suppression

Suppressing OSPF Forwarding Address in Translated Type-5 LSAs Example

This example suppresses the forwarding address in translated Type-5 LSAs:

```
interface gigabitethernet 0/0/0
 ip address 10.93.1.1 255.255.255.0
 ip ospf cost 1
!
interface gigabitethernet 0/01
 ip address 10.94.1.1 255.255.255.0
!
router ospf 1
 network 10.93.0.0 0.0.255.255 area 0.0.0.0
 network 10.94.0.0 0.0.255.255 area 10
 area 10 nssa translate type7 suppress-fa
```

Additional References

The following sections provide references related to OSPF Forwarding Address Suppression in Translated Type-5 LSAs:

Related Documents

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF	"Configuring OSPF"
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases
OSPFv3 Address Families	" *OSPFv3 Address Families*" module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1587	*The OSPF NSSA Option* **Note** Configuring the OSPF Forwarding Address Suppression in Translated Type-5 LSAs feature causes the router to be noncompliant with RFC 1587, *The OSPF NSSA Option* .

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Forwarding Address Suppression

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 238: Feature Information for OSPF Forwarding Address Suppression in Translated Type-5 LSAs

Feature Name	Releases	Feature Information
OSPF Forwarding Address Suppression in Translated Type-5 LSAs	Cisco IOS XE Release 2.1	The OSPF Forwarding Address Suppression in Translated Type-5 LSAs feature causes a not-so-stubby area (NSSA) area border router (ABR) to translate Type-7 link state advertisements (LSAs) to Type-5 LSAs, but to use the address 0.0.0.0 for the forwarding address instead of that specified in the Type-7 LSA. This feature causes routers that are configured not to advertise forwarding addresses into the backbone to direct forwarded traffic to the translating NSSA ABRs. The following commands are introduced or modified in the feature documented in this module: • **area nssa translate** • **show ip ospf**

Table 239: Feature Information for OSPF Forwarding Address Suppression in Translated Type-5 LSAs

Feature Name	Releases	Feature Information
OSPF Forwarding Address Suppression in Translated Type-5 LSAs	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 213

OSPF Inbound Filtering Using Route Maps with a Distribute List

The OSPF Inbound Filtering Using Route Maps with a Distribute List feature allows users to define a route map to prevent Open Shortest Path First (OSPF) routes from being added to the routing table. In the route map, the user can match on any attribute of the OSPF route.

Prerequisites OSPF Inbound Filtering Using Route Maps with a Distribute List

It is presumed that you have OSPF configured in your network.

Information About OSPF Inbound Filtering Using Route Maps with a Distribute List

Benefits of OSPF Route-Map-Based-Filtering

Users can define a route map to prevent OSPF routes from being added to the routing table. This filtering happens at the moment when OSPF is installing the route in the routing table. This feature has no effect on LSA flooding. In the route map, the user can match on any attribute of the OSPF route. That is, the route map could be based on the following **match** options:

- **match interface**
- **match ip address**

- **match ip next-hop**
- **match ip route-source**
- **match metric**
- **match route-type**
- **match tag**

This feature can be useful during redistribution if the user tags prefixes when they get redistributed on ASBRs and later uses the tag to filter the prefixes from being installed in the routing table on other routers.

Filtering Based on Route Tag

Users can assign tags to external routes when they are redistributed to OSPF. Then the user can deny or permit those routes in the OSPF domain by identifying that tag in the **route-map** and **distribute-list in** commands.

Filtering Based on Route Type

In OSPF, the external routes could be Type 1 or Type 2. Users can create route maps to match either Type 1 or Type 2 and then use the **distribute-list in** command to filter certain prefixes. Also, route maps can identify internal routes (interarea and intra-area) and then those routes can be filtered.

Filtering Based on Route Source

When a match is done on the route source, the route source represents the OSPF Router ID of the LSA originator of the LSA in which the prefix is advertised.

Filtering Based on Interface

When a match is done on the interface, the interface represents the outgoing interface for the route that OSPF is trying to install in the routing table.

Filtering Based on Next-Hop

When a match is done on the next hop, the next hop represents the next hop for the route that OSPF is trying to install in the routing table.

How to Configure OSPF Inbound Filtering Using Route Maps

Configuring OSPF Inbound Filtering Using a Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match tag** *tag-name*
5. Repeat Steps 3 and 4 with other **route-map** and **match** commands if you choose.

6. **exit**
7. **router ospf** *process-id*
8. **distribute-list route-map** *map-tag* **in**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map tag-filter deny 10`	Defines a route map to control filtering.

Step 4	**match tag** *tag-name*
	Example:
	Example:
	`or other match commands`
	Example:
	`Router(config-router)# match tag 777`
		Matches routes with a specified name, to be used as the route map is referenced.
		• At least one **match** command is required, but it need not be this **match**command. This is just an example.
		• The list of **match** commands available to be used in this type of route map appears on the **distribute-list in** command reference page.
		• This type of route map will have no **set** commands.

Step 5	Repeat Steps 3 and 4 with other **route-map** and **match** commands if you choose.	--
Step 6	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode.
Step 7	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an OSPF routing process.
Step 8	**distribute-list route-map** *map-tag* **in** **Example:**	Enables filtering based on an OSPF route map.

	Command or Action	Purpose
	`Router(config-router)# distribute-list route-map tag-filter in`	
Step 9	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode.

Configuration Examples for OSPF Inbound Filtering Using Route Maps with a Distribute List

Example OSPF Route-Map-Based Filtering

In this example, OSPF external LSAs have a tag. The value of the tag is examined before the prefix is installed in the routing table. All OSPF external prefixes that have the tag value of 777 are filtered (prevented from being installed in the routing table). The permit statement with sequence number 20 has no match conditions, and there are no other route-map statements after sequence number 20, so all other conditions are permitted.

```
route-map tag-filter deny 10
 match tag 777
route-map tag-filter permit 20
!
router ospf 1
 router-id 10.0.0.2
 log-adjacency-changes
 network 172.16.2.1 0.0.0.255 area 0
 distribute-list route-map tag-filter in
```

Additional References

The following sections provide references related to configuring the OSPF Inbound Filtering Using Route Maps with a Distribute List feature.

Related Documents

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF	"Configuring OSPF"
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Inbound Filtering Using Route Maps with a Distribute List

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 240: Feature Information for OSPF Inbound Filtering Using Route Maps with a Distribute List

Feature Name	Releases	Feature Information
OSPF Inbound Filtering Using Route Maps with a Distribute List	Cisco IOS XE Release 2.1	The OSPF Inbound Filtering Using Route Maps with a Distribute List feature allows users to define a route map to prevent OSPF routes from being added to the routing table. The following commands are introduced or modified in the feature documented in this module: • **distribute-list in (IP)**

Table 241: Feature Information for OSPF Inbound Filtering Using Route Maps with a Distribute List

Feature Name	Releases	Feature Information
OSPF Inbound Filtering Using Route Maps with a Distribute List	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 214

OSPFv3 Route Filtering Using Distribute-List

The OSPFv3 route filtering using distribute-list feature allows users to filter the incoming routes that are programmed in routing table, and the outgoing routes that are advertised.

Prerequisites for OSPFv3 Route Filtering Using Distribute-List

It is presumed that you have OSPF configured in your network.

Information About OSPFv3 Route Filtering Using Distribute-List

Users can define a route map to prevent OSPF routes from being added to the routing table. This filtering happens at the moment when OSPF is installing the route in the routing table. This feature has no effect on link-state advertisement (LSA) flooding.

This feature can be useful during redistribution if the user tags prefixes when they get redistributed on Autonomous System Boundary Routers (ASBRs) and later uses the tag to filter the prefixes from being installed in the routing table on other routers. The below mentioned options are available only for distribute-list filtering using route-map.

Filtering Based on Route Tag

Users can assign tags to external routes when they are redistributed to OSPF. Then the user can deny or permit those routes in the OSPF domain by identifying that tag in the **route-map** and **distribute-list in** or **distribute-list out** commands.

Filtering Based on Route Type

In OSPF, the external routes could be Type 1 or Type 2. Users can create route maps to match either Type 1 or Type 2 and then use the **distribute-list in** command to filter certain prefixes. Also, route maps can identify internal routes (interarea and intra-area) and then those routes can be filtered.

Filtering Based on Route Source

When a match is done on the route source, the route source represents the OSPF Router ID of the LSA originator of the LSA in which the prefix is advertised.

Filtering Based on Interface

When a match is done on the interface, the interface represents the outgoing interface for the route that OSPF is trying to install in the routing table.

Filtering Based on Next Hop

When a match is done on the next hop, the next hop represents the next hop for the route that OSPF is trying to install in the routing table.

Note The **distribute-list in** command can be configured to prevent routes from being installed in the global Routing Information Base (RIB). Prior to the implementation of OSPF local RIB (for feature information on OSPF local RIB, see OSPFv2 Local RIB), OSPF would attempt to install a less preferred route (e.g. an inter-area route when the intra-area path is filtered). With OSPF local RIB, only the best route is considered (because this is the only route the local RIB maintains). There is no concept of a "second-best" OSPF route. For more information on the routing algorithm used by Cisco OSPF routers, please refer to RFC 2328.

How to Configure OSPFv3 Route Filtering Using Distribute-List

Configuring OSPFv3 (IPv4 address-family)

Command Mode: Address family mode (address-family ipv4 unicast). Following is the syntax:

```
[no] distribute-list [<access-list #> | <access-list name>] |
                     {prefix <name1> gateway <name2>} |
                     {prefix <name1>} | {gateway <name2>} |
       {route-map name} in [<interface>]

[no] distribute-list [<access-list #> | <access-list name>] | [prefix <name>] out
      [{ <routing-process> | <interface> }]
```

Interface: Incoming (used with Inbound filtering) or outgoing (used with outbound filtering) interface.

Routing-process: Source protocol for the route to be filtered.

Configuring Inbound Filtering: Route Map

SUMMARY STEPS

1. Configure OSPFv3.
2. Configure address-family ipv4 unicast.
3. Configure distribute list with the appropriate route-map.

DETAILED STEPS

Step 1 Configure OSPFv3.

```
Device(config)#router ospfv3 1
```

Step 2 Configure address-family ipv4 unicast.

```
Device(config-router)#address-family ipv4 unicast
```

Step 3 Configure distribute list with the appropriate route-map.

```
Device(config-router-af)#distribute-list route-map rmap-name in
```

The following match options in a route-map are supported:

- match interface
- match ip address
- match ip next-hop
- match ip route-source
- match metric
- match route-type
- match tag

Configuring Inbound Filtering: Prefix-List/Access-List

SUMMARY STEPS

1. Configure OSPFv3.
2. Configure address-family ipv4 unicast.
3. Defines prefix list to be used and the direction for the filter.

DETAILED STEPS

Step 1 Configure OSPFv3.

```
Device(config)#router ospfv3 1
```

Step 2 Configure address-family ipv4 unicast.

```
Device(config-router)#address-family ipv4 unicast
```

Step 3 Defines prefix list to be used and the direction for the filter.

```
Device(config-router-af)#distribute-list prefix pfxname in
```

Note The following are the available optional arguments. You can use these arguments to filter based on incoming interface. Choose any interface that is available on your device.

```
Ethernet           IEEE 802.3
  Loopback           Loopback interface
  Null               Null interface
  Port-channel       Ethernet Channel of interfaces
  Serial             Serial
  Tunnel             Tunnel interface
  Vlan               Catalyst Vlans
```

Configuring Outbound Filtering

SUMMARY STEPS

1. Configure OSPFv3.
2. Configure address-family ipv4 unicast.
3. Configure distribute list with the appropriate route-map.

DETAILED STEPS

Step 1 Configure OSPFv3.

```
Device(config)#router ospfv3 1
```

Step 2 Configure address-family ipv4 unicast.

```
Device(config-router)#address-family ipv4 unicast
```

Step 3 Configure distribute list with the appropriate route-map.

```
Device(config-router-af)#distribute-list prefix pfxlist-name out
```

Note The following are the available optional arguments. You can use these options to filter based on the source protocol of the route.

```
bgp        Border Gateway Protocol (BGP)
  connected  Connected
  eigrp      Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis       ISO IS-IS
  lisp       Locator ID Separation Protocol (LISP)
  ospf       Open Shortest Path First (OSPF)
  ospfv3     OSPFv3
  rip        Routing Information Protocol (RIP)
  static     Static routes
```

Configuring Route Filtering Using Distribute-List for OSPFv3 (IPv6 address-family)

Mode: Address-family mode (address-family ipv6 unicast). Prefix-list and route-map are supported as filtering options. Following is the syntax:

```
[no] distribute-list prefix-list <name> in [<interface>]
[no] distribute-list route-map <name> in
[no] distribute-list prefix-list <name> out <routing-process>
```

Interface: Incoming (used with Inbound filtering) or outgoing (used with outbound filtering) interface.

Routing-process: Source protocol for the route to be filtered.

Configuring Inbound Filtering: Route Map

SUMMARY STEPS

1. Configure OSPFv3.
2. Configure address-family ipv6unicast.
3. Define route map.

DETAILED STEPS

Step 1 Configure OSPFv3.

```
Device(config)#router ospfv3 1
```

Step 2 Configure address-family ipv6unicast.

```
Device(config-router)#address-family ipv6 unicast
```

Step 3 Define route map.

```
Device(config-router-af)#distribute-list route-map rmap-name in
```

The following match options in a route-map are supported:

- match interface
- match ip address
- match ip next-hop
- match metric
- match route-type
- match tag

Configuring Inbound Filtering: Prefix-List

SUMMARY STEPS

1. Configure OSPFv3.
2. Configure address-family ipv6 unicast.
3. Define prefix list name.
4. Define filter incoming routing updates.

DETAILED STEPS

Step 1 Configure OSPFv3.

```
Device(config)#router ospfv3 1
```

Step 2 Configure address-family ipv6 unicast.

```
Device(config-router)#address-family ipv6 unicast
```

Step 3 Define prefix list name.

```
Device(config-router-af)#distribute-list prefix pfxlist-name
```

Step 4 Define filter incoming routing updates.

```
Device(config-router-af)#distribute-list prefix pfxname in
```

Note The following are the available optional arguments. You can use these arguments to filter based on incoming interface. Choose any interface that is available on your device.

```
Ethernet            IEEE 802.3
  Loopback            Loopback interface
  Null                Null interface
  Port-channel        Ethernet Channel of interfaces
  Serial              Serial
  Tunnel              Tunnel interface
  Vlan                Catalyst Vlans
```

Configuring Outbound Filtering

SUMMARY STEPS

1. Configure OSPFv3.
2. Configure address-family ipv6 unicast.
3. Define prefix list name.

DETAILED STEPS

Step 1 Configure OSPFv3.

```
Device(config)#router ospfv3 1
```

Step 2 Configure address-family ipv6 unicast.

```
Device(config-router)#address-family ipv6 unicast
```

Step 3 Define prefix list name.

```
Device(config-router-af)#distribute-list prefix-list pfxlist-name out
```

Note These are the available options for the routing process. The <**routing-process**> argument is mandatory for IPv6 outbound route filtering.

```
bgp         Border Gateway Protocol (BGP)
  connected  Connected Routes
```

```
eigrp       Enhanced Interior Gateway Routing Protocol (EIGRP)
isis        ISO IS-IS
lisp        Locator ID Separation Protocol (LISP)
ospf        Open Shortest Path First (OSPFv3)
rip         IPv6 Routing Information Protocol (RIPv6)
static      Static Routes
```

Additional References

Related Documents

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 Route Filtering Using Distribute-List

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 242: Feature Information for OSPFv3 Route Filtering Using Distribute-List

Feature Name	Releases	Feature Information
OSPFv3 Route Filtering Using Distribute-List	Cisco IOS XE Denali 16.3.1	The route-map support for OSPFv3 route-filtering using distribute-list is supported.

Table 243: Feature Information for OSPFv3 Route Filtering Using Distribute-List

Feature Name	Releases	Feature Information
OSPFv3 Route Filtering Using Distribute-List	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 215

OSPF Shortest Path First Throttling

The OSPF Shortest Path First Throttling feature makes it possible to configure shortest path first (SPF) scheduling in millisecond intervals and to potentially delay SPF calculations during network instability. SPF is scheduled to calculate the Shortest Path Tree (SPT) when there is a change in topology. One SPF run may include multiple topology change events.

The interval at which the SPF calculations occur is chosen dynamically and is based on the frequency of topology changes in the network. The chosen interval is within the boundary of the user-specified value ranges. If the network topology is unstable, SPF throttling calculates SPF scheduling intervals to be longer until the topology becomes stable.

Information About OSPF SPF Throttling

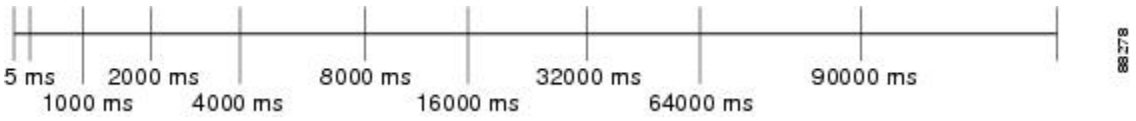
SPF calculations occur at the interval set by the **timers throttle spf**command. The wait interval indicates the amount of time to wait until the next SPF calculation occurs. Each wait interval after that calculation is twice as long as the previous one until the wait interval reaches the maximum wait time specified.

The SPF timing can be better explained using an example. In this example the start interval is set at 5 milliseconds (ms), the wait interval at 1000 milliseconds, and the maximum wait time is set at 90,000 milliseconds.

```
timers throttle spf 5 1000 90000
```

The figure below shows the intervals at which the SPF calculations occur so long as at least one topology change event is received in a given wait interval.

Figure 210: SPF Calculation Intervals Set by the timers throttle spf Command

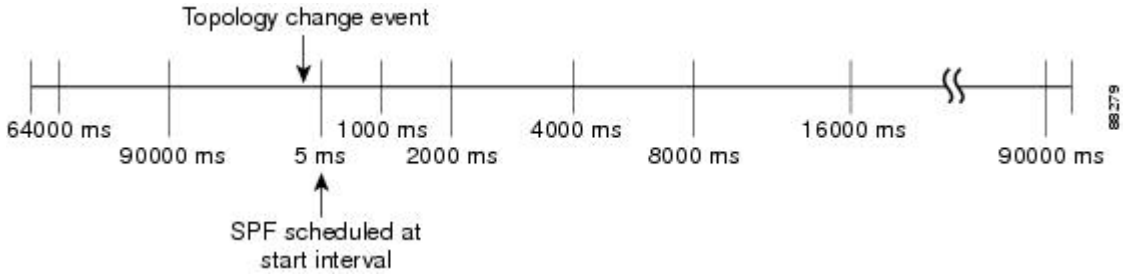


Notice that the wait interval between SPF calculations doubles when at least one topology change event is received during the previous wait interval. Once the maximum wait time is reached, the wait interval remains the same until the topology stabilizes and no event is received in that interval.

If the first topology change event is received after the current wait interval, the SPF calculation is delayed by the amount of time specified as the start interval. The subsequent wait intervals continue to follow the dynamic pattern.

If the first topology change event occurs after the maximum wait interval begins, the SPF calculation is again scheduled at the start interval and subsequent wait intervals are reset according the parameters specified in the **timers throttle spf**command. Notice in the figure below that a topology change event was received after the start of the maximum wait time interval and that the SPF intervals have been reset.

Figure 211: Timer Intervals Reset After a Topology Change Event



How to Configure OSPF SPF Throttling

Configuring OSPF SPF Throttling

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **timers throttle spf** *spf-start spf-hold spf-max-wait*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an OSPF routing process.
Step 4	**timers throttle spf** *spf-start spf-hold spf-max-wait* **Example:** `Router(config-router)# timers throttle spf 10 4800 90000`	Sets OSPF throttling timers.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits configuration mode.

Verifying SPF Throttle Values

To verify SPF throttle timer values, use the **show ip ospf** command. The values are displayed in the lines that begin, "Initial SPF schedule delay...," "Minimum hold time between two consecutive SPFs...," and "Maximum wait time between two consecutive SPFs...."

```
Router# show ip ospf
 Routing Process "ospf 1" with ID 10.10.10.2 and Domain ID 0.0.0.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 It is an autonomous system boundary router
 Redistributing External Routes from,
    static, includes subnets in redistribution
Initial SPF schedule delay 5 msecs
 Minimum hold time between two consecutive SPFs 1000 msecs
 Maximum wait time between two consecutive SPFs 90000 msecs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 4. Checksum Sum 0x17445
 Number of opaque AS LSA 0. Checksum Sum 0x0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        Area has no authentication
        SPF algorithm last executed 19:11:15.140 ago
        SPF algorithm executed 28 times
        Area ranges are
        Number of LSA 4. Checksum Sum 0x2C1D4
        Number of opaque link LSA 0. Checksum Sum 0x0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

Configuration Example for OSPF SPF Throttling

Example Throttle Timers

This example shows a router configured with the start, hold, and maximum interval values for the **timers throttle spf** command set at 5, 1,000, and 90,000 milliseconds, respectively.

```
router ospf 1
 router-id 10.10.10.2
 log-adjacency-changes
 timers throttle spf 5 1000 90000
 redistribute static subnets
 network 21.21.21.0 0.0.0.255 area 0
 network 22.22.22.0 0.0.0.255 area 00
```

Additional References

The following sections provide references related to OSPF Shortest Path First Throttling.

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Shortest Path First Throttling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 244: Feature Information for OSPF Shortest Path First Throttling

Feature Name	Releases	Feature Information
OSPF Shortest Path First Throttling	Cisco IOS XE Release 2.1	The OSPF Shortest Path First Throttling feature makes it possible to configure SPF scheduling in millisecond intervals and to potentially delay shortest path first (SPF) calculations during network instability. SPF is scheduled to calculate the Shortest Path Tree (SPT) when there is a change in topology. The following commands are introduced or modified in the feature documented in this module: • **timer spf-interval** • **timers throttle spf**

Table 245: Feature Information for OSPF Shortest Path First Throttling

Feature Name	Releases	Feature Information
OSPF Shortest Path First Throttling	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 216

OSPF Support for Fast Hello Packets

The OSPF Support for Fast Hello Packets feature provides a way to configure the sending of hello packets in intervals less than 1 second. Such a configuration results in faster convergence in an Open Shortest Path First (OSPF) network.

Note It is recommended to use Bidirectional Forwarding Detection (BFD) instead of Fast Hello Packets.

Prerequisites for OSPF Support for Fast Hello Packets

OSPF must be already configured in the network or must be configured at the same time as the OSPF Support for Fast Hello Packets feature.

Information About OSPF Support for Fast Hello Packets

OSPF Hello Interval and Dead Interval

OSPF hello packets are packets that an OSPF process sends to its OSPF neighbors to maintain connectivity with those neighbors. The hello packets are sent at a configurable interval (in seconds). The defaults are 10 seconds for an Ethernet link and 30 seconds for a non broadcast link. Hello packets include a list of all neighbors for which a hello packet has been received within the dead interval. The dead interval is also a configurable interval (in seconds), and defaults to four times the value of the hello interval. The value of all hello intervals must be the same within a network. Likewise, the value of all dead intervals must be the same within a network.

These two intervals work together to maintain connectivity by indicating that the link is operational. If a router does not receive a hello packet from a neighbor within the dead interval, it will declare that neighbor to be down.

OSPF Fast Hello Packets

OSPF fast hello packets refer to hello packets being sent at intervals of less than 1 second. To understand fast hello packets, you should already understand the relationship between OSPF hello packets and the dead interval. See the section OSPF Hello Interval and Dead Interval, on page 2741.

OSPF fast hello packets are achieved by using the **ip ospf dead-interval** command. The dead interval is set to 1 second, and the hello-multiplier value is set to the number of hello packets you want to send during that 1 second, thus providing subsecond or "fast" hello packets.

When fast hello packets are configured on the interface, the hello interval advertised in the hello packets that are sent out this interface is set to 0. The hello interval in the hello packets received over this interface is ignored.

The dead interval must be consistent on a segment, whether it is set to 1 second (for fast hello packets) or set to any other value. The hello multiplier need not be the same for the entire segment as long as at least one hello packet is sent within the dead interval.

Benefits of OSPF Fast Hello Packets

The benefit of the OSPF Support for Fast Hello Packets feature is that your OSPF network will experience faster convergence time than it would without fast hello packets. This feature allows you to detect lost neighbors within 1 second. It is especially useful in LAN segments, where neighbor loss might not be detected by the Open System Interconnection (OSI) physical layer and data-link layer.

How to Configure OSPF Fast Hello Packets

Configuring OSPF Fast Hello Packets

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf dead-interval minimal hello-multiplier** *multiplier*
5. **end**
6. **show ip ospf interface** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip ospf dead-interval minimal hello-multiplier** *multiplier* **Example:** `Router(config-if)# ip ospf dead-interval minimal hello-multiplier 5`	Sets the interval during which at least one hello packet must be received, or else the neighbor is considered down. • In the example, OSPF Support for Fast Hello Packets is enabled by specifying the **minimal** keyword and the **hello-multiplier** keyword and value. Because the multiplier is set to 5, five hello packets will be sent every second.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Saves configuration commands to the running configuration file, exits configuration mode, and returns to privileged EXEC mode. • Use this command when you are ready to exit configuration mode and save the configuration to the running configuration file.
Step 6	**show ip ospf interface** [*interface-type interface-number*] **Example:** `Router# show ip ospf interface gigabitethernet 0/0/1`	(Optional) Displays OSPF-related interface information. • The relevant fields that verify OSPF fast hello packets are indicated in the sample output following this table.

Examples

The following sample output verifies that OSPF Support for Fast Hello Packets is configured. In the line that begins with "Timer intervals configured," the hello interval is 200 milliseconds, the dead interval is 1 second, and the next hello packet is due in 76 milliseconds.

```
Router# show ip ospf interface gigabitethernet 0/0/1
GigabitEthernet0/0/1 is up, line protocol is up
  Internet Address 172.16.1.2/24, Area 0
  Process ID 1, Router ID 172.17.0.2, Network Type BROADCAST, Cost:1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.17.0.2, Interface address 172.16.1.2
  Backup Designated router (ID) 172.16.0.1, Interface address 172.16.1.1
  Timer intervals configured, Hello 200 msec, Dead 1, Wait 1, Retransmit 5
    Hello due in 76 msec
Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
```

```
Last flood scan length is 2, maximum is 3
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 172.16.0.1  (Backup Designated Router)
Suppress hello for 0 neighbor(s)
```

Configuration Examples for OSPF Support for Fast Hello Packets

Example OSPF Fast Hello Packets

The following example configures OSPF fast hello packets; the dead interval is 1 second and 5 hello packets are sent every second:

```
interface gigabitethernet 0/0/1
 ip ospf dead-interval minimal hello-multiplier 5
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
OSPFv3 External Path Preference Option	"*Configuring OSPF*" module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Support for Fast Hello Packets

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 246: Feature Information for OSPF Support for Fast Hello Packets

Feature Name	Releases	Feature Information
OSPF Support for Fast Hello Packets	Cisco IOS XE Release 2.1	The OSPF Support for Fast Hello Packets feature provides a way to configure the sending of hello packets in intervals less than 1 second. Such a configuration results in faster convergence in an Open Shortest Path First (OSPF) network.

Table 247: Feature Information for OSPF Support for Fast Hello Packets

Feature Name	Releases	Feature Information
OSPF Support for Fast Hello Packets	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 217

OSPF Incremental SPF

The Open Shortest Path First (OSPF) protocol can be configured to use an incremental SPF algorithm for calculating the shortest path first routes. Incremental SPF is more efficient than the full SPF algorithm, thereby allowing OSPF to converge faster on a new routing topology in reaction to a network event.

Prerequisites for OSPF Incremental SPF

It is presumed that you have OSPF configured in your network.

Information About OSPF Incremental SPF

OSPF uses Dijkstra's SPF algorithm to compute the shortest path tree (SPT). During the computation of the SPT, the shortest path to each node is discovered. The topology tree is used to populate the routing table with routes to IP networks. When changes to a Type-1 or Type-2 link-state advertisement (LSA) occur in an area, the entire SPT is recomputed. In many cases, the entire SPT need not be recomputed because most of the tree remains unchanged. Incremental SPF allows the system to recompute only the affected part of the tree. Recomputing only a portion of the tree rather than the entire tree results in faster OSPF convergence and saves CPU resources. Note that if the change to a Type-1 or Type-2 LSA occurs in the calculating router itself, then the full SPT is performed.

Incremental SPF is scheduled in the same way as the full SPF. Routers enabled with incremental SPF and routers not enabled with incremental SPF can function in the same internetwork.

How to Enable OSPF Incremental SPF

Enabling Incremental SPF

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **ispf**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an OSPF routing process.
Step 4	**ispf** **Example:** `Router(config-router)# ispf`	Enables incremental SPF.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode.

Configuration Examples for OSPF Incremental SPF

Example Incremental SPF

This example enables incremental SPF:

```
router ospf 1
 ispf
```

Additional References

The following sections provide references related to OSPF Incremental SPF.

Related Documents

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF	"Configuring OSPF"
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Incremental SPF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 248: Feature Information for OSPF Incremental SPF

Feature Name	Releases	Feature Information
OSPF Incremental SPF	Cisco IOS XE Release 2.1	OSPF can be configured to use an incremental SPF algorithm for calculating the shortest path first routes. Incremental SPF is more efficient than the full SPF algorithm, thereby allowing OSPF to converge faster on a new routing topology in reaction to a network event The following commands are introduced or modified in the feature documented in this module: • **ispf**

Table 249: Feature Information for OSPF Incremental SPF

Feature Name	Releases	Feature Information
OSPF Incremental SPF	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 218

OSPF Limit on Number of Redistributed Routes

Open Shortest Path First (OSPF) supports a user-defined maximum number of prefixes (routes) that are allowed to be redistributed into OSPF from other protocols or other OSPF processes. Such a limit could help prevent the router from being flooded by too many redistributed routes.

Prerequisites for OSPF Limit on Number of Redistributed Routes

It is presumed that you have OSPF configured in your network, along with another protocol or another OSPF process you are redistributing.

Information About OSPF Limit on Number of Redistributed Routes

If large number of IP routes are sent into OSPF by redistributing Border Gateway Protocol (BGP) into OSPF, the network can be severely flooded. Limiting the number of redistributed routes prevents this potential problem.

OSPF can receive and accept packets from non-routable addresses (for example, 0.0.0.0/7) also.

How to Limit the Number of OSPF Redistributed Routes

This section contains the following procedures, which are mutually exclusive. That is, you cannot both limit redistributed prefixes and also choose to be warned.

Limiting the Number of Redistributed Routes

Note You cannot both limit redistributed prefixes and also choose to be warned.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **redistribute** *protocol* [*process-id* | *as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match**{**internal**| **external 1**| **external 2**}][**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
5. **redistribute maximum-prefix** *maximum* [*threshold*]
6. **end**
7. **show ip ospf** [*process-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an OSPF routing process.
Step 4	**redistribute** *protocol* [*process-id*	*as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match**{**internal**| **external 1**| **external 2**}][**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Router(config-router)# redistribute eigrp 10`	Redistributes routes from one routing domain into another routing domain.
Step 5	**redistribute maximum-prefix** *maximum* [*threshold*] **Example:** `Router(config-router)# redistribute maximum-prefix 100 80`	Sets a maximum number of IP prefixes that are allowed to be redistributed into OSPF. • There is no default value for the *maximum* argument. • The *threshold* value defaults to 75 percent.

	Command or Action	Purpose
		Note If the **warning-only** keyword had been configured in this command, no limit would be enforced; a warning message is simply logged.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode.
Step 7	**show ip ospf** [*process-id*] **Example:** `Router# show ip ospf 1`	(Optional) Displays general information about OSPF routing processes. • If a redistribution limit was configured, the output will include the maximum limit of redistributed prefixes and the threshold for warning messages.

Requesting a Warning About the Number of Routes Redistributed into OSPF

Note You cannot both limit redistributed prefixes and also choose to be warned.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **redistribute** *protocol* [*process-id* | *as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match**{**internal**| **external 1**| **external 2**}][**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
5. **redistribute maximum-prefix** *maximum* [*threshold*] **warning-only**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an OSPF routing process.
Step 4	**redistribute** *protocol* [*process-id* | *as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match**{**internal**| **external 1**| **external 2**}][**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Router(config-router)# redistribute eigrp 10`	Redistributes routes from one routing domain into another routing domain.
Step 5	**redistribute maximum-prefix** *maximum* [*threshold*] **warning-only** **Example:** `Router(config-router)# redistribute maximum-prefix 1000 80 warning-only`	Causes a warning message to be logged when the maximum number of IP prefixes has been redistributed into OSPF. • Because the **warning-only** keyword is included, no limit is imposed on the number of redistributed prefixes into OSPF. • There is no default value for the *maximum* argument. • The *threshold* value defaults to 75 percent. • This example causes two warnings: one at 80 percent of 1000 (800 routes redistributed) and another at 1000 routes redistributed.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode.

Configuration Examples for OSPF Limit on Number of Redistributed Routes

Example OSPF Limit the Number of Redistributed Routes

This example sets a maximum of 1200 prefixes that can be redistributed into OSPF process 1. Prior to reaching the limit, when the number of prefixes redistributed reaches 80 percent of 1200 (960 prefixes), a warning message is logged. Another warning is logged when the limit is reached and no more routes are redistributed.

```
router ospf 1
 router-id 10.0.0.1
 domain-id 5.6.7.8
 log-adjacency-changes
```

```
 timers lsa-interval 2
 network 10.0.0.1 0.0.0.0 area 0
 network 10.1.5.1 0.0.0.0 area 0
 network 10.2.2.1 0.0.0.0 area 0
 redistribute static subnets
 redistribute maximum-prefix 1200 80
```

Example Requesting a Warning About the Number of Redistributed Routes

This example allows two warning messages to be logged, the first if the number of prefixes redistributed reaches 85 percent of 600 (510 prefixes), and the second if the number of redistributed routes reaches 600. However, the number of redistributed routes is not limited.

```
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
 redistribute eigrp 10 subnets
 redistribute maximum-prefix 600 85 warning-only
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
OSPFv3 Address Families	*OSPFv3 Address Families* module

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 5187.	*OSPFv3 Graceful Restart*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Limit on Number of Redistributed Routes

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 250: Feature Information for OSPF Limit on Number of Redistributed Routes

Feature Name	Releases	Feature Information
OSPF Limit on Number of Redistributed Routes	Cisco IOS XE Release 2.1 Cisco IOS XE Release 2.6	OSPF supports a user-defined maximum number of prefixes (routes) that are allowed to be redistributed into OSPF from other protocols or other OSPF processes. Such a limit could help prevent the router from being flooded by too many redistributed routes. The following commands are introduced or modified in the feature documented in this module: • **redistribute maximum-prefix** • **show ip ospf** • **show ip ospf database**

Table 251: Feature Information for OSPF Limit on Number of Redistributed Routes

Feature Name	Releases	Feature Information
OSPF Limit on Number of Redistributed Routes	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 219

OSPFv3 Fast Convergence: LSA and SPF Throttling

The Open Shortest Path First version 3 (OSPFv3) link-state advertisement (LSAs) and shortest-path first (SPF) throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPFv3 during times of network instability. It also allows faster OSPFv3 convergence by providing LSA rate limiting in milliseconds.

Information About OSPFv3 Fast Convergence: LSA and SPF Throttling

Fast Convergence: LSA and SPF Throttling

The OSPFv3 LSA and SPF throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPFv3 during times of network instability. It also allows faster OSPFv3 convergence by providing LSA rate limiting in milliseconds.

OSPFv3 can use static timers for rate-limiting SPF calculation and LSA generation. Although these timers are configurable, the values used are specified in seconds, which poses a limitation on OSPFv3 convergence. LSA and SPF throttling achieves subsecond convergence by providing a more sophisticated SPF and LSA rate-limiting mechanism that is able to react quickly to changes and also provide stability and protection during prolonged periods of instability.

How to Configure OSPFv3 Fast Convergence: LSA and SPF Throttling

Tuning LSA and SPF Timers for OSPFv3 Fast Convergence

This task can be performed in Cisco IOS Release 15.1(3)S and 15.2(1)T and later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **timers lsa arrival** *milliseconds*
5. **timers pacing flood** *milliseconds*
6. **timers pacing lsa-group** *seconds*
7. **timers pacing retransmission** *milliseconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**timers lsa arrival** *milliseconds* **Example:** `Device(config-rtr)# timers lsa arrival 300`	Sets the minimum interval at which the software accepts the same LSA from OSPFv3 neighbors.
Step 5	**timers pacing flood** *milliseconds* **Example:** `Device(config-rtr)# timers pacing flood 30`	Configures LSA flood packet pacing.

	Command or Action	Purpose
Step 6	**timers pacing lsa-group** *seconds* **Example:** `Device(config-router)# timers pacing lsa-group 300`	Changes the interval at which OSPFv3 LSAs are collected into a group and refreshed, checksummed, or aged.
Step 7	**timers pacing retransmission** *milliseconds* **Example:** `Device(config-router)# timers pacing retransmission 100`	Configures LSA retransmission packet pacing in IPv4 OSPFv3.

Configuring LSA and SPF Throttling for OSPFv3 Fast Convergence

This task can be performed in releases prior to Cisco IOS Release 15.1(3)S and 15.2(1)T.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **timers throttle spf** *spf-start spf-hold spf-max-wait*
5. **timers throttle lsa** *start-interval hold-interval max-interval*
6. **timers lsa arrival** *milliseconds*
7. **timers pacing flood** *milliseconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router ospf** *process-id* **Example:** `Device(config)# ipv6 router ospf 1`	Enables OSPFv3 router configuration mode.
Step 4	**timers throttle spf** *spf-start spf-hold spf-max-wait* **Example:**	Turns on SPF throttling.

	Command or Action	Purpose
	`Device(config-rtr)# timers throttle spf 200 200 200`	
Step 5	**timers throttle lsa** *start-interval hold-interval max-interval* **Example:** `Device(config-rtr)# timers throttle lsa 300 300 300`	Sets rate-limiting values for OSPFv3 LSA generation.
Step 6	**timers lsa arrival** *milliseconds* **Example:** `Device(config-rtr)# timers lsa arrival 300`	Sets the minimum interval at which the software accepts the same LSA from OSPFv3 neighbors.
Step 7	**timers pacing flood** *milliseconds* **Example:** `Device(config-rtr)# timers pacing flood 30`	Configures LSA flood packet pacing.

Configuration Examples for OSPFv3 Fast Convergence: LSA and SPF Throttling

Example: Configuring LSA and SPF Throttling for OSPFv3 Fast Convergence

The following example show how to display the configuration values for SPF and LSA throttling timers:

```
Device# show ipv6 ospf

Routing Process "ospfv3 1" with ID 10.9.4.1
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 It is an autonomous system boundary router
 Redistributing External Routes from,
    ospf 2
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*
OSPFv3 Fast Convergence: LSA and SPF Throttling	"*OSPF Link-State Advertisement Throttling* " module

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	IPv6 RFCs

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 Fast Convergence: LSA and SPF Throttling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 252: Feature Information for OSPFv3 Fast Convergence: LSA and SPF Throttling

Feature Name	Releases	Feature Information
OSPFv3 Fast Convergence: LSA and SPF Throttling	Cisco IOS XE Release 2.1	The OSPFv3 LSA and SPF throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPFv3 during times of network instability.

Table 253: Feature Information for OSPFv3 Fast Convergence: LSA and SPF Throttling

Feature Name	Releases	Feature Information
OSPFv3 Fast Convergence: LSA and SPF Throttling	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 220

OSPFv3 Max-Metric Router LSA

The Open Shortest Path First version 3 (OSPFv3) max-metric router link-state advertisement (LSA) feature enables OSPFv3 to advertise its locally generated router LSAs with a maximum metric. The feature allows OSPFv3 processes to converge but not attract transit traffic through the device if there are better alternate paths.

Information About OSPFv3 Max-Metric Router LSA

OSPFv3 Max-Metric Router LSA

The OSPFv3 max-metric router LSA feature enables OSPFv3 to advertise its locally generated router LSAs with a maximum metric. The feature allows OSPFv3 processes to converge but not attract transit traffic through the device if there are better alternate paths. After a specified timeout or a notification from Border Gateway Protocol (BGP), OSPFv3 advertises the LSAs with normal metrics.

The max-metric LSA control places the OSPFv3 router into the stub router role using its LSA advertisement. A stub router only forwards packets destined to go to its directly connected links. In OSPFv3 networks, a device could become a stub router by advertising large metrics for its connected links, so that the cost of a path through this device becomes larger than that of an alternative path. OSPFv3 stub router advertisement allows a device to advertise the infinity metric (0xFFFF) for its connected links in router LSAs and advertise the normal interface cost if the link is a stub network.

How to Configure OSPFv3 Max-Metric Router LSA

Configuring the OSPFv3 Max-Metric Router LSA

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** *process-id*
4. **address-family ipv6 unicast**
5. **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**inter-area-lsas** [*max-metric-value*]] [**on-startup** {*seconds* | **wait-for-bgp**}] [**prefix-lsa**] [**stub-prefix-lsa** [*max-metric-value*]] [**summary-lsa** [*max-metric-value*]]
6. **end**
7. **show ospfv3** [*process-id*] **max-metric**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** *process-id* **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode.
Step 4	**address-family ipv6 unicast** **Example:** `Device(config)# address-family ipv6 unicast`	Configures an instance of the OSPFv3 process in the IPv6 address family.
Step 5	**max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**inter-area-lsas** [*max-metric-value*]] [**on-startup** {*seconds* | **wait-for-bgp**}] [**prefix-lsa**] [**stub-prefix-lsa** [*max-metric-value*]] [**summary-lsa** [*max-metric-value*]] **Example:**	Configures a device that is running the OSPFv3 protocol to advertise a maximum metric so that other devices do not prefer the device as an intermediate hop in their SPF calculations.

	Command or Action	Purpose
	`Device(config-router-af)# max-metric router-lsa on-startup wait-for-bgp`	
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.
Step 7	**show ospfv3** [*process-id*] **max-metric** **Example:** `Device# show ospfv3 1 max-metric`	Displays OSPFv3 maximum metric origination information.

Configuration Examples for OSPFv3 Max-Metric Router LSA

Example: Verifying the OSPFv3 Max-Metric Router LSA

```
Router# show ipv6 ospf max-metric

            OSPFv3 Router with ID (192.1.1.1) (Process ID 1)

 Start time: 00:00:05.886, Time elapsed: 3d02h
 Originating router-LSAs with maximum metric
    Condition: always, State: active
```

Additional References for OSPF Nonstop Routing

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
Configuring IETF NSF or Cisco NSF	"Configuring NSF-OSPF" module in the *Cisco IOS High Availability Configuration Guide*

Standard and RFCs

Standard/RFC	Title
RFC 2328	*OSPF Version 2*

Standard/RFC	Title
RFC 3623	*Graceful OSPF Restart*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 Max-Metric Router LSA

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 254: Feature Information for OSPFv3 Max-Metric Router LSA

Feature Name	Releases	Feature Information
OSPFv3 Max-Metric Router LSA	Cisco IOS XE Release 3.4S	The OSPFv3 max-metric router LSA feature enables OSPF to advertise its locally generated router LSAs with a maximum metric. The following commands were introduced or modified: **max-metric router-lsa**, **show ipv6 ospf max-metric**, **show ospfv3 max-metric**.

Table 255: Feature Information for OSPFv3 Max-Metric Router LSA

Feature Name	Releases	Feature Information
OSPFv3 Max-Metric Router LSA	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 221

OSPF Link-State Advertisement Throttling

The OSPF Link-State Advertisement Throttling feature provides a dynamic mechanism to slow down link-state advertisement (LSA) updates in Open Shortest Path First (OSPF) during times of network instability. It also allows faster OSPF convergence by providing LSA rate limiting in milliseconds.

Prerequisites for OSPF LSA Throttling

It is presumed that you have OSPF configured in your network.

Information About OSPF LSA Throttling

Benefits of OSPF LSA Throttling

Prior to the OSPF LSA Throttling feature, LSA generation was rate-limited for 5 seconds. That meant that changes in an LSA could not be propagated in milliseconds, so the OSPF network could not achieve millisecond convergence.

The OSPF LSA Throttling feature is enabled by default and allows faster OSPF convergence (in milliseconds). This feature can be customized. One command controls the generation (sending) of LSAs, and another command controls the receiving interval. This feature also provides a dynamic mechanism to slow down the frequency of LSA updates in OSPF during times of network instability.

How OSPF LSA Throttling Works

The **timers throttle lsa all** command controls the generation (sending) of LSAs. The first LSA is always generated immediately upon an OSPF topology change, and the next LSA generated is controlled by the minimum start interval. The subsequent LSAs generated for the same LSA are rate-limited until the maximum

interval is reached. The "same LSA" is defined as an LSA instance that contains the same LSA ID number, LSA type, and advertising router ID.

The **timers lsa arrival** command controls the minimum interval for accepting the same LSA. If an instance of the same LSA arrives sooner than the interval that is set, the LSA is dropped. It is recommended that the arrival interval be less than or equal to the hold-time interval of the **timers throttle lsa all** command.

How to Customize OSPF LSA Throttling

Customizing OSPF LSA Throttling

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **timers throttle lsa all** *start-interval hold-interval max-interval*
5. **timers lsa arrival** *milliseconds*
6. **end**
7. **show ip ospf timers rate-limit**
8. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an OSPF routing process.
Step 4	**timers throttle lsa all** *start-interval hold-interval max-interval* **Example:** `Router(config-router)# timers throttle lsa all 100 10000 45000`	(Optional) Sets the rate-limiting values (in milliseconds) for LSA generation. • The default values are as follows: • *start-interval*is 0 milliseconds. • *hold-interval*is 5000 milliseconds.

	Command or Action	Purpose
		• *max-interval*is 5000 milliseconds.
Step 5	**timers lsa arrival** *milliseconds* **Example:** `Router(config-router)# timers lsa arrival 2000`	(Optional) Sets the minimum interval (in milliseconds) between instances of receiving the same LSA. • The default value is 1000 milliseconds. • We suggest you keep the *milliseconds*value of the LSA arrival timer less than or equal to the neighbors' *hold-interval* value of the **timers throttle lsa all** command.
Step 6	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode.

Step 7	**show ip ospf timers rate-limit**
	Example:
	`Router# show ip ospf timers rate-limit`
	Example:
	Example:
	`LSAID: 10.1.1.1 Type: 1 Adv Rtr: 172.16.2.2 Due in: 00:00:00.028`
	Example:
	`LSAID: 192.168.4.1 Type: 3 Adv Rtr: 172.17.2.2 Due in: 00:00:00.028`
		(Optional) Displays a list of the LSAs in the rate limit queue (about to be generated).
		• The example shows two LSAs in the queue. Each LSA is identified by LSA ID number, Type (of LSA), Advertising router ID, and the time in hours:minutes:seconds (to the milliseconds) when the LSA is due to be generated.

Step 8	**show ip ospf**
	Example:
	`Router# show ip ospf`
	Example:
	Example:
	`Routing Process "ospf 4" with ID 10.10.24.4`
	Example:
	`Supports only single TOS(TOS0) routes`
	Example:
		(Optional) Displays information about OSPF.
		• The output lines that specify initial throttle delay, minimum hold time for LSA throttle, and maximum wait time for LSA throttle indicate the LSA throttling values.

	Command or Action	Purpose
	Supports opaque LSA **Example:** Supports Link-local Signaling (LLS) **Example:** Initial SPF schedule delay 5000 msecs **Example:** Minimum hold time between two consecutive SPFs 10000 msecs **Example:** Maximum wait time between two consecutive SPFs 10000 msecs **Example:** Incremental-SPF disabled **Example:** Initial LSA throttle delay 100 msecs **Example:** Minimum hold time for LSA throttle 10000 msecs **Example:** Maximum wait time for LSA throttle 45000 msecs **Example:** Minimum LSA arrival 1000 msecs **Example:** LSA group pacing timer 240 secs **Example:** Interface flood pacing timer 33 msecs **Example:** Retransmission pacing timer 66 msecs **Example:** Number of external LSA 0. Checksum Sum 0x0 **Example:**	

	Command or Action	Purpose
	Number of opaque AS LSA 0. Checksum Sum 0x0 **Example:** Number of DCbitless external and opaque AS LSA 0 **Example:** Number of DoNotAge external and opaque AS LSA 0 **Example:** Number of areas in this router is 1. 1 normal 0 stub 0 nssa **Example:** External flood list length 0 **Example:** Area 24 **Example:** Number of interfaces in this area is 2 **Example:** Area has no authentication **Example:** SPF algorithm last executed 04:28:18.396 ago **Example:** SPF algorithm executed 8 times **Example:** Area ranges are **Example:** Number of LSA 4. Checksum Sum 0x23EB9 **Example:** Number of opaque link LSA 0. Checksum Sum 0x0 **Example:** Number of DCbitless LSA 0 **Example:**	

	Command or Action	Purpose
	Number of indication LSA 0 **Example:** Number of DoNotAge LSA 0 **Example:** Flood list length 0	

Configuration Examples for OSPF LSA Throttling

Example OSPF LSA Throttling

This example customizes OSPF LSA throttling so that the start interval is 200 milliseconds, the hold interval is 10,000 milliseconds, and the maximum interval is 45,000 milliseconds. The minimum interval between instances of receiving the same LSA is 2000 milliseconds.

```
router ospf 1
 log-adjacency-changes
 timers throttle lsa all 200 10000 45000
 timers lsa arrival 2000
 network 10.10.4.0 0.0.0.255 area 24
 network 10.10.24.0 0.0.0.255 area 24
```

Additional References

The following sections provide references related to OSPF LSA throttling.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF	"Configuring OSPF"
OSPFv3 Fast Convergence: LSA and SPF Throttling	"*OSPFv3 Fast Convergence: LSA and SPF Throttling*" module
OSPFv3 Max-Metric Router LSA	"*OSPFv3 Max-Metric Router LSA*" module

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Link-State Advertisement Throttling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 256: Feature Information for OSPF Link-State Advertisement Throttling

Feature Name	Releases	Feature Information
OSPF Link-State Advertisement Throttling	Cisco IOS XE Release 2.1 Cisco IOS XE Release 2.6	The OSPF Link-State Advertisement Throttling feature provides a dynamic mechanism to slow down link-state advertisement (LSA) updates in OSPF during times of network instability. It also allows faster OSPF convergence by providing LSA rate limiting in milliseconds. The following commands are introduced or modified in the feature documented in this module: • **debug ip ospf database-timer rate-limit** • **show ip ospf** • **show ip ospf timers rate-limit** • **timers lsa arrival** • **timers throttle lsa all**

Table 257: Feature Information for OSPF Link-State Advertisement Throttling

Feature Name	Releases	Feature Information
OSPF Link-State Advertisement Throttling	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 222

OSPF Support for Unlimited Software VRFs per PE Router

In a Multiprotocol Label Switching--Virtual Private Network (MPLS-VPN) deployment, each VPN routing and forwarding instance (VRF) needs a separate Open Shortest Path First (OSPF) process when configured to run OSPF. The OSPF Support for Unlimited Software VRFs per Provider Edge (PE) Router feature addresses the scalability issue for OSPF VPNs by eliminating the OSPF VPN limit of 32 processes.

Prerequisites for OSPF Support for Unlimited Software VRFs per PE Router

You must have OSPF configured in your network.

Restrictions for OSPF Support for Unlimited Software VRFs per PE Router

Only 32 processes per VRF can be supported. For different VRF processes, there is no limit.

Information About OSPF Support for Unlimited Software VRFs per PE Router

Before Cisco IOS Releases 12.3(4)T and 12.0(27)S, a separate OSPF process was necessary for each VRF that receives VPN routes via OSPF. When VPNs are deployed, an MPLS Provider Edge (PE) router will be running both multiprotocol Border Gateway Protocol (BGP) for VPN distribution, and Interior Gateway Protocol (IGP) for PE-P connectivity. OSPF is commonly used as the IGP between a customer edge (CE) router and a PE router. OSPF was not scalable in a VPN deployment because of the limit of 32 processes. By default, one process is used for connected routes and another process is used for static routes; therefore only 28 processes can be created for VRFs.

The OSPF Support for Unlimited Software VRFs per Provider Edge Router feature allows for an approximate range of 300 to 10,000 VRFs, depending on the particular platform and on the applications, processes, and protocols that are currently running on the platform.

How to Configure OSPF Support for Unlimited Software VRFs per PE Router

Configuring Unlimited Software VRFs per PE Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vpn-name*
4. **exit**
5. **router ospf** *process-id* [**vrf** *vpn-name*]
6. **end**
7. **show ip ospf** [*process-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip vrf** *vpn-name* **Example:** `Router(config)# ip vrf crf-1`	Defines a VPN routing and forwarding (VRF) instance and enters VRP configuration mode.
Step 4	**exit** **Example:** `Router(config-vrf)# exit`	Returns to global configuration mode.
Step 5	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Router(config)# router ospf 1 vrf crf-1`	Enables OSPF routing. • The *process-id* argument identifies the OSPF process. • Use the **vrf** keyword and *vpn-name* argument to identify the VPN already defined in Step 3. **Note** You can now configure as many OSPF VRF processes as needed. Repeat Steps 3-5 as needed.
Step 6	**end** **Example:** `Router(config-router)# end`	Returns to privileged EXEC mode.
Step 7	**show ip ospf** [*process-id*] **Example:** `Router# show ip ospf 1`	Displays general information about OSPF routing processes.

Configuration Examples for OSPF Support for Unlimited Software VRFs per PE Router

Example Configuring OSPF Support for Unlimited Software VRFs per PE Router

This example shows a basic OSPF configuration using the **router ospf** command to configure OSPF VRF processes for the VRFs first, second, and third:

```
Router> enable
Router# configure terminal
Router(config)# ip vrf first
Router(config-vrf)# exit
Router(config)# ip vrf second
Router(config-vrf)# exit
Router(config)# ip vrf third
Router(config-vrf)# exit
```

```
Router(config)# router ospf 12 vrf first
Router(config-router)# exit
Router(config)# router ospf 13 vrf second
Router(config-router)# exit
Router(config)# router ospf 14 vrf third
Router(config)# end
```

Example Verifying OSPF Support for Unlimited Software VRFs per PE Router

This example illustrates the output from the **show ip ospf** command to verify that OSPF VRF process 12 has been created for the VRF named first. The output that relates to the VRF first appears in bold.

```
Router# show ip ospf 12
main ID type 0x0005, value 0.0.0.100
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Connected to MPLS VPN Superbackbone, VRF first
 It is an area border router
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x0
 Number of opaque AS LSA 0. Checksum Sum 0x0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm last executed 00:00:15.204 ago
        SPF algorithm executed 2 times
        Area ranges are
        Number of LSA 1. Checksum Sum 0xD9F3
        Number of opaque link LSA 0. Checksum Sum 0x0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

Additional References

The following sections provide references related to the OSPF Support for Unlimited Software VRFs per Provider Edge Router feature.

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Support for Unlimited Software VRFs per PE Router

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 258: Feature Information for OSPF Support for Unlimited Software VRFs per Provider Edge Router

Feature Name	Releases	Feature Information
OSPF Support for Unlimited Software VRFs per Provider Edge Router	Cisco IOS XE Release 2.1	In a Multiprotocol Label Switching--Virtual Private Network (MPLS-VPN) deployment, each VPN routing and forwarding instance (VRF) needs a separate Open Shortest Path First (OSPF) process when configured to run OSPF. The OSPF Support for Unlimited Software VRFs per Provider Edge Router feature addresses the scalability issue for OSPF VPNs by eliminating the OSPF VPN limit of 32 processes.

Table 259: Feature Information for OSPF Support for Unlimited Software VRFs per Provider Edge Router

Feature Name	Releases	Feature Information
OSPF Support for Unlimited Software VRFs per Provider Edge Router	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 223

OSPF Area Transit Capability

The OSPF Area Transit Capability feature provides an OSPF Area Border Router (ABR) with the ability to discover shorter paths through the transit area for forwarding traffic that would normally need to travel through the virtual-link path. This functionality allows Cisco IOS XE software to be compliant with RFC 2328, *OSPF Version 2*.

Information About OSPF Area Transit Capability

How the OSPF Area Transit Capability Feature Works

The OSPF Area Transit Capability feature is enabled by default. RFC 2328 defines OSPF area transit capability as the ability of the area to carry data traffic that neither originates nor terminates in the area itself. This capability enables the OSPF ABR to discover shorter paths through the transit area and to forward traffic along those paths rather than using the virtual link or path, which is not optimal.

For a detailed description of OSPF area transit capability, see RFC 2328, OSPF Version 2 .

How to Disable OSPF Area Transit Capability

Disabling OSPF Area Transit Capability on an Area Border Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **no capability transit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Router(config)# router ospf 100`	Enables OSPF routing and enters router configuration mode. • The *process-id* argument identifies the OSPF process.
Step 4	**no capability transit** **Example:** `Router(config-router)# no capability transit`	Disables OSPF area transit capability on all areas for a router process.

Additional References

The following sections provide references related to the OSPF Area Transit Capability feature.

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2328	OSPF Version 2

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Area Transit Capability

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 260: Feature Information for OSPF Area Transit Capability

Feature Name	Releases	Feature Information
OSPF Area Transit Capability	Cisco IOS XE Release 2.1	The OSPF Area Transit Capability feature provides an OSPF Area Border Router (ABR) the ability to discover shorter paths through the transit area for forwarding traffic that would normally need to travel through the virtual-link path. This functionality allows Cisco IOS XE software to be compliant with RFC 2328. The command related to this feature is • **capability transit**

Table 261: Feature Information for OSPF Area Transit Capability

Feature Name	Releases	Feature Information
OSPF Area Transit Capability	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 224

OSPF Per-Interface Link-Local Signaling

The OSPF Per-Interface Link-Local Signaling feature allows you to selectively enable or disable Link-Local Signaling (LLS) for a specific interface regardless of the global (router level) setting that you have previously configured.

Information About OSPF Per-Interface Link-Local Signaling

LLS allows for the extension of existing OSPF packets in order to provide additional bit space. The additional bit space enables greater information per packet exchange between OSPF neighbors. This functionality is used, for example, by the OSPF Nonstop Forwarding (NSF) Awareness feature that allows customer premises equipment (CPE) routers that are NSF-aware to help NSF-capable routers perform nonstop forwarding of packets.

When LLS is enabled at the router level, it is automatically enabled for all interfaces. The OSPF Per-Interface Link-Local Signaling feature allows you to selectively enable or disable LLS for a specific interface. You may want to disable LLS on a per-interface basis depending on your network design. For example, disabling LLS on an interface that is connected to a non-Cisco device that may be noncompliant with RFC 2328 can prevent problems with the forming of OSPF neighbors in the network.

How to Configure OSPF Per-Interface Link-Local Signaling

Turning Off LLS on a Per-Interface Basis

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot /port*

4. **ip address** *ip-address mask* [**secondary**]
5. **no ip directed-broadcast** [*access-list-number* | *extended access-list-number*]
6. **ip ospf message-digest-key** *key-id encryption-type* **md5** *key*
7. [**no** | **default**] **ip ospf lls** [**disable**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot /port* **Example:** `Router(config)# interface gigabitethernet 1/1/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.2.145.20 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**no ip directed-broadcast** [*access-list-number* | *extended access-list-number*] **Example:** `Router(config-if)# no ip directed-broadcast`	Drops directed broadcasts destined for the subnet to which that interface is attached, rather than broadcasting them. • The forwarding of IP directed broadcasts on Ethernet interface 1/0 is disabled.
Step 6	**ip ospf message-digest-key** *key-id encryption-type* **md5** *key* **Example:** `Router(config-if)# ip ospf message-digest-key 100 md5 testing`	Enables OSPF Message Digest 5 (MD5) algorithm authentication.
Step 7	[**no** | **default**] **ip ospf lls** [**disable**] **Example:** `Router(config-if)# ip ospf lls disable`	Disables LLS on an interface, regardless of the global (router level) setting.

What to Do Next

To verify that LLS has been enabled or disabled for a specific interface, use the **show ip ospf interface** command. See the "Example: Configuring and Verifying the OSPF Per-Interface Link-Local Signaling Feature" section for an example of the information displayed.

Configuration Examples for OSPF Per-Interface Link-Local Signaling

Example Configuring and Verifying OSPF Per-Interface Link-Local Signaling

In the following example, LLS has been enabled on GigabitEthernet interface 1/1/0 and disabled on GigabitEthernet interface 2/1/0:

```
interface gigabitethernet1/1/0
 ip address 10.2.145.2 255.255.255.0
 no ip directed-broadcast
 ip ospf message-digest-key 1 md5 testing
 ip ospf lls
!
interface gigabitethernet2/1/0
 ip address 10.1.145.2 255.255.0.0
 no ip directed-broadcast
 ip ospf message-digest-key 1 md5 testing
!
 ip ospf lls disable
interface Ethernet3/0
 ip address 10.3.145.2 255.255.255.0
 no ip directed-broadcast
!
router ospf 1
 log-adjacency-changes detail
 area 0 authentication message-digest
 redistribute connected subnets
 network 10.0.0.0 0.255.255.255 area 1
 network 10.2.3.0 0.0.0.255 area 1
```

In the following example, the **show ip ospf interface** command has been entered to verify that LLS has been enabled for GigabitEthernet interface 1/1/0 and disabled for GigabitEthernet interface 2/1/0:

```
Router# show ip ospf interface
GigabitEthernet1/1/0 is up, line protocol is up
  Internet Address 10.2.145.2/24, Area 1
  Process ID 1, Router ID 10.22.222.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.2.2.3, Interface address 10.2.145.1
  Backup Designated router (ID) 10.22.222.2, Interface address 10.2.145.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
!   Supports Link-local Signaling (LLS)
Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 2, maximum is 8
  Last flood scan time is 0 msec, maximum is 0 msec
```

```
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.2.2.3  (Designated Router)
  Suppress hello for 0 neighbor(s)
GigabitEthernet2/1/0 is up, line protocol is up
  Internet Address 10.1.145.2/16, Area 1
  Process ID 1, Router ID 10.22.222.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.2.2.3, Interface address 10.1.145.1
  Backup Designated router (ID) 10.22.222.2, Interface address 10.1.145.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:04
!   Does not support Link-local Signaling (LLS)
Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 2, maximum is 11
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 45.2.2.3  (Designated Router)
    Suppress hello for 0 neighbor(s)
GigabitEthernet3/1/0 is up, line protocol is up
  Internet Address 10.3.145.2/24, Area 1
  Process ID 1, Router ID 10.22.222.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.2.2.3, Interface address 10.3.145.1
  Backup Designated router (ID) 10.22.222.2, Interface address 10.3.145.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:07
!   Supports Link-local Signaling (LLS)
Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 2, maximum is 11
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.2.2.3  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

Additional References

The following sections provide references related to the OSPF Per-Interface Link-Local Signaling feature.

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
Configuring OSPF NSF Awareness	"Cisco Nonstop Forwarding"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2328	*OSPF Version 2*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Per-Interface Link-Local Signaling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 262: Feature Information for OSPF Per-Interface Link-Local Signaling

Feature Name	Releases	Feature Information
OSPF Per-Interface Link-Local Signaling	Cisco IOS XE Release 2.1	The OSPF Per-Interface Link-Local Signaling feature allows you to selectively enable or disable Link-Local Signaling (LLS) for a specific interface regardless of the global (router level) setting that you have previously configured. The following commands are introduced or modified in the feature documented in this module: • **ip ospf lls**

Table 263: Feature Information for OSPF Per-Interface Link-Local Signaling

Feature Name	Releases	Feature Information
OSPF Per-Interface Link-Local Signaling	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 225

OSPF Link-State Database Overload Protection

The OSPF Link-State Database Overload Protection feature allows you to limit the number of nonself-generated link-state advertisements (LSAs) for a given Open Shortest Path First (OSPF) process. Excessive LSAs generated by other routers in the OSPF domain can substantially drain the CPU and memory resources of the router.

Prerequisites for OSPF Link-State Database Overload Protection

It is presumed that you have OSPF running on your network.

Information About OSPF Link-State Database Overload Protection

Benefits of Using OSPF Link-State Database Overload Protection

The OSPF Link-State Database Overload Protection feature provides a mechanism at the OSPF level to limit the number of nonself-generated LSAs for a given OSPF process. When other routers in the network have been misconfigured, they may generate a high volume of LSAs, for instance, to redistribute large numbers of prefixes. This protection mechanism prevents routers from receiving a large number of LSAs and therefore experiencing CPU and memory shortages.

How OSPF Link-State Database Overload Protection Works

When the OSPF Link-State Database Overload Protection feature is enabled, the router keeps a count of the number of received (nonself-generated) LSAs that it has received. When the configured threshold number of LSAs is reached, an error message is logged. When the configured maximum number of LSAs is exceeded,

the router will send a notification. If the count of received LSAs is still higher than the configured maximum after one minute, the OSPF process takes down all adjacencies and clears the OSPF database. In this ignore state, all OSPF packets received on any interface that belongs to this OSPF process are ignored and no OSPF packets are generated on any of these interfaces. The OSPF process remains in the ignore state for the time configured by the **ignore-time** keyword of the **max-lsa** command. Each time the OSPF process gets into an ignore state a counter is incremented. If this counter exceeds the number of minutes configured by the **ignore-count** keyword, the OSPF process stays permanently in the same ignore state and manual intervention is required to get the OSPF process out of the ignore state. The ignore state counter is reset to 0 when the OSPF process remains in the normal state of operation for the amount of time that was specified by the **reset-time** keyword.

If the **warning-only** keyword of the **max-lsa** command has been configured, the OSPF process will send only a warning that the LSA maximum has been exceeded.

How to Configure OSPF Link-State Database Overload Protection

Limiting the Number of Self-Generating LSAs for an OSPF Process

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **router-id** *ip-address*
5. **log -adjacency-changes** [**detail**]
6. **max-lsa** *maximum-number* [*threshold-percentage*] [**warning-only**] [**ignore-time** *minutes*] [**ignore-count** *count-number*] [**reset-time** *minutes*]
7. **network** *ip-address wildcard-mask* **area** *area-id*
8. **end**
9. **show ip ospf** [*process-id area-id*] **database**[**database-summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:**	Enables OSPF routing. • The *process-id* argument identifies the OSPF process.

	Command or Action	Purpose
	`Router(config)# router ospf 1`	
Step 4	**router-id** *ip-address* **Example:** `Router(config-router)# router-id 10.0.0.1`	Specifies a fixed router ID for an OSPF process.
Step 5	**log -adjacency-changes** [**detail**] **Example:** `Router(config-router)# log-adjacency-changes`	Configures the router to send a syslog message when an OSPF neighbor goes up or down.
Step 6	**max-lsa** *maximum-number* [*threshold-percentage*] [**warning-only**] [**ignore-time** *minutes*] [**ignore-count** *count-number*] [**reset-time** *minutes*] **Example:** `Router(config-router)# max-lsa 12000`	Limits the number of nonself-generated LSAs that an OSPF routing process can keep in the OSPF link-state database (LSDB).
Step 7	**network** *ip-address wildcard-mask* **area** *area-id* **Example:** `Router(config-router)# network 209.165.201.1 255.255.255.255 area 0`	Defines the interfaces on which OSPF runs and defines the area ID for those interfaces.
Step 8	**end** **Example:** `Router(config-router)# end`	Ends the current configuration mode and returns to Privileged EXEC mode.
Step 9	**show ip ospf** [*process-id area-id*] **database**[**database-summary**] **Example:** `Router# show ip ospf 2000 database database-summary`	Displays lists of information related to the OSPF database for a specific router. • Use this command to verify the number of nonself-generated LSAs on a router.

Example

The **show ip ospf** command is entered with the **database-summary** keyword to verify the actual number of nonself-generated LSAs on a router. This command can be used at any time to display lists of information related to the OSPF database for a specific router.

```
Router# show ip ospf 2000 database database-summary

            OSPF Router with ID (192.168.1.3) (Process ID 2000)
Area 0 database summary
  LSA Type      Count    Delete   Maxage
  Router        5        0        0
```

```
  Network       2         0         0
  Summary Net   8         2         2
  Summary ASBR  0         0         0
  Type-7 Ext    0         0         0
    Prefixes redistributed in Type-7  0
  Opaque Link   0         0         0
  Opaque Area   0         0         0
  Subtotal      15        2         2
Process 2000 database summary
  LSA Type      Count     Delete    Maxage
  Router        5         0         0
  Network       2         0         0
  Summary Net   8         2         2
  Summary ASBR  0         0         0
  Type-7 Ext    0         0         0
  Opaque Link   0         0         0
  Opaque Area   0         0         0
  Type-5 Ext    4         0         0
    Prefixes redistributed in Type-5  0
  Opaque AS     0         0         0
  Non-self      16
  Total         19        2         2
```

Configuration Examples for OSPF Link-State Database Overload Protection

Setting a Limit for LSA Generation Example

In the following example, the router is configured to not accept any more nonself-generated LSAs once a maximum of 14,000 has been exceeded:

```
Router(config)# router ospf 1
Router(config-router)# router-id 192.168.0.1
Router(config-router)# log-adjacency-changes
Router(config-router)# max-lsa 14000
Router(config-router)# area 33 nssa
Router(config-router)# network 192.168.0.1 0.0.0.0 area 1
Router(config-router)# network 192.168.5.1 0.0.0.0 area 1
Router(config-router)# network 192.168.2.1 0.0.0.0 area 0
```

In the following example, the **show ip ospf** command has been entered to confirm the configuration:

```
Router# show ip ospf 1
Routing Process "ospf 1" with ID 192.168.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Maximum number of non self-generated LSA allowed 14000
   Threshold for warning message 75%
   Ignore-time 5 minutes, reset-time 10 minutes
   Ignore-count allowed 5, current ignore-count 0
It is an area border and autonomous system boundary router
```

In the following example, the following output appears when the **show ip ospf** command has been entered during the time when the router is in the ignore state:

```
Router# show ip ospf 1
Routing Process "ospf 1" with ID 192.168.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Maximum number of non self-generated LSA allowed 14000
   Threshold for warning message 75%
   Ignore-time 5 minutes, reset-time 10 minutes
   Ignore-count allowed 5, current ignore-count 1
   Ignoring all neighbors due to max-lsa limit, time remaining: 00:04:52
It is an area border and autonomous system boundary router
```

The following output appears when the **show ip ospf** command has been entered after the router left the ignore state:

```
Router# show ip ospf 1
Routing Process "ospf 1" with ID 192.168.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Maximum number of non self-generated LSA allowed 14000
   Threshold for warning message 75%
   Ignore-time 5 minutes, reset-time 10 minutes
   Ignore-count allowed 5, current ignore-count 1 - time remaining: 00:09:51
It is an area border and autonomous system boundary router
```

The following output appears when the **show ip ospf** command has been entered for a router that is permanently in the ignore state:

```
Router# show ip ospf 1
Routing Process "ospf 1" with ID 192.168.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Maximum number of non self-generated LSA allowed 14000
   Threshold for warning message 75%
   Ignore-time 5 minutes, reset-time 10 minutes
   Ignore-count allowed 5, current ignore-count 6
   Permanently ignoring all neighbors due to max-lsa limit
It is an area border and autonomous system boundary router
```

Additional References

The following sections provide references related to the OSPF Link-State Database Overload Protection feature.

Related Documents

Related Topic	Document Title
Configuring OSPF	" Configuring OSPF"

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Link-State Database Overload Protection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 226

OSPF MIB Support of RFC 1850 and Latest Extensions

The OSPF MIB Support of RFC 1850 and Latest Extensions feature introduces the capability for Simple Network Management Protocol (SNMP) monitoring on the Open Shortest Path First (OSPF) routing protocol. Users have an improved ability to constantly monitor the changing state of an OSPF network by use of MIB objects to gather information relating to protocol parameters and trap notification objects that can signal the occurrence of significant network events such as transition state changes. The protocol information collected by the OSPF MIB objects and trap objects can be used to derive statistics that will help monitor and improve overall network performance.

Prerequisites for OSPF MIB Support of RFC 1850 and Latest Extensions

- OSPF must be configured on the router.
- Simple Network Management Protocol (SNMP) must be enabled on the router before notifications (traps) can be configured or before SNMP GET operations can be performed.

Information About OSPF MIB Support of RFC 1850 and Latest Extensions

The following sections contain information about MIB objects standardized as part of RFC 1850 and defined in OSPF-MIB and OSPF-TRAP-MIB. In addition, extensions to RFC 1850 objects are described as defined in the two Cisco private MIBs, CISCO-OSPF-MIB and CISCO-OSPF-TRAP-MIB.

OSPF MIB Changes to Support RFC 1850

OSPF MIB

This section describes the new MIB objects that are provided by RFC 1850 definitions. These OSPF MIB definitions provide additional capacity that is not provided by the standard OSPF MIB that supported the previous RFC 1253. To see a complete set of OSPF MIB objects, see the OSPF-MIB file.

The table below shows the new OSPF-MIB objects that are provided by RFC 1850 definitions. The objects are listed in the order in which they appear within the OSPF-MIB file, per the tables that describe them.

Table 264: New OSPF-MIB Objects

OSPF-MIB Table	New MIB Objects
OspfAreaEntry table	• OspfAreaSummary • OspfAreaStatus
OspfStubAreaEntry	• OspfStubMetricType
OspfAreaRangeEntry	• OspfAreaRangeEffect
OspfHostEntry	• OspfHostAreaID
OspfIfEntry	• OspfIfStatus • OspfIfMulticastForwarding • OspfIfDemand • OspfIfAuthType
OspfVirtIfEntry	• OspfVirtIfAuthType
OspfNbrEntry	• OspfNbmaNbrPermanence • OspfNbrHelloSuppressed
OspfVirtNbrEntry	• OspfVirtNbrHelloSuppressed

OSPF-MIB Table	New MIB Objects
OspfExtLsdbEntry	• OspfExtLsdbType • OspfExtLsdbLsid • OspfExtLsdbRouterId • OspfExtLsdbSequence • OspfExtLsdbAge • OspfExtLsdbChecksum • OspfExtLsdbAdvertisement
OspfAreaAggregateEntry	• OspfAreaAggregateAreaID • OspfAreaAggregateLsdbType • OspfAreaAggregateNet • OspfAreaAggregateMask • OspfAreaAggregateStatusospfSetTrap • OspfAreaAggregateEffect

OSPF TRAP MIB

This section describes scalar objects and MIB objects that are provided to support RFC 1850.

The following scalar objects are added to OSPF-TRAP-MIB and are listed in the order in which they appear in the OSPF-TRAP-MIB file:

- OspfExtLsdbLimit
- OspfMulticastExtensions
- OspfExitOverflowInterval
- OspfDemandExtensions

The ospfSetTrap control MIB object contains the OSPF trap MIB objects that enable and disable OSPF traps in the IOS CLI. These OSPF trap MIB objects are provided by the RFC 1850 standard OSPF MIB. To learn how to enable and disable the OSPF traps, see the How to Enable OSPF MIB Support of RFC 1850 and Latest Extensions, on page 2807.

The table below shows the OSPF trap MIB objects, listed in the order in which they appear within the OSPF-TRAP-MIB file.

Table 265: New OSPF-TRAP-MIB Objects

OSPF Control MIB Object	Trap MIB Objects
ospfSetTrap	• ospfIfStateChange • ospfVirtIfStateChange • ospfNbrStateChange • ospfVirtNbrState • ospfIfConfigError • ospfVirtIfConfigError • ospfIfAuthFailure • ospfVirtIfAuthFailure • ospfIfRxBadPacket • ospfVirtIfRxBadPacket • ospfTxRetransmit • ospfVirtIfTxRetransmit • ospfOriginateLsa • ospfMaxAgeLsa

CISCO OSPF MIB

This section describes scalar and Cisco-specific OSPF MIB objects that are provided as extensions to support the RFC 1850 OSPF MIB definitions, to provide capability that the standard MIB cannot provide.

The following scalar objects are added to OSPF-OSPF-MIB:

- cospfRFC1583Compatibility
- cospfOpaqueLsaSupport
- cospfOpaqueASLsaCount
- cospfOpaqueASLsaCksumSum

For each of the following table entries, the new Cisco-specific MIB objects that are provided as extensions to support the RFC 1850 OSPF MIB definitions are listed. To see the complete set of objects for the Cisco-specific OSPF MIB, refer to the CISCO-OSPF-MIB file.

The table below shows the new CISCO-OSPF-MIB objects that are provided by RFC 1850 definitions. The objects are listed in the order in which they appear within the CISCO-OSPF-MIB file, per the tables that describe them.

Table 266: New CISCO-OSPF-MIB Objects

CISCO-OSPF-MIB Table	New MIB Objects
cospfAreaEntry	• cospfOpaqueAreaLsaCount • cospfOpaqueAreaLsaCksumSum • cospfAreaNssaTranslatorRole • cospfAreaNssaTranslatorState • cospfAreaNssaTranslatorEvents
cospfLsdbEntry	• cospfLsdbType • cospfLsdbSequence • cospfLsdbAge • cospfLsdbChecksum • cospfLsdbAdvertisement
cospfIfEntry	• cospfIfLsaCount • cospfIfLsaCksumSum
cospfVirtIfEntry	• cospfVirtIfLsaCount • cospfVirtIfLsaCksumSum
cospfLocalLsdbEntry	• cospfLocalLsdbIpAddress • cospfLocalLsdbAddressLessIf • cospfLocalLsdbType • cospfLocalLsdbLsid • cospfLocalLsdbRouterId • cospfLocalLsdbSequence • cospfLocalLsdbAge • cospfLocalLsdbChecksum • cospfLocalLsdbAdvertisement

CISCO-OSPF-MIB Table	New MIB Objects
cospfVirtLocalLsdbEntry	• cospfVirtLocalLsdbTransitArea • cospfVirtLocalLsdbNeighbor • cospfVirtLocalLsdbType • cospfVirtLocalLsdbLsid • cospfVirtLocalLsdbRouterId • cospfVirtLocalLsdbSequence • cospfVirtLocalLsdbAge • cospfVirtLocalLsdbChecksum • cospfVirtLocalLsdbAdvertisement

CISCO OSPF TRAP MIB

The cospfSetTrap MIB object represents trap events in CISCO-OSPF-TRAP-MIB. This is a bit map, where the first bit represents the first trap. The following MIB objects are TRAP events that have been added to support RFC 1850. To see a complete set of Cisco OSPF Trap MIB objects, see the CISCO-OSPF-TRAP-MIB file.

The table below shows the trap events described within the cospfSetTrap MIB object in the CISCO-TRAP-MIB:

Table 267: CISCO-OSPF Trap Events

CISCO-OSPF-TRAP-MIB Trap Events	Trap Event Description
cospfIfConfigError	This trap is generated for mismatched MTU parameter errors that occur when nonvirtual OSPF neighbors are forming adjacencies.
cospfVirtIfConfigError	This trap is generated for mismatched MTU parameter errors when virtual OSPF neighbors are forming adjacencies.
cospfTxRetransmit	This trap is generated in the case of opaque LSAs when packets are sent by a nonvirtual interface. An opaque link-state advertisement (LSA) is used in MPLS traffic engineering to distribute attributes such as capacity and topology of links in a network. The scope of this LSA can be confined to the local network (Type 9, Link-Local), OSPF area (Type 20, Area-Local), or autonomous system (Type 11, AS scope). The information in an opaque LSA can be used by an external application across the OSPF network.
cospfVirtIfTxRetransmit	This trap is generated in the case of opaque LSAs when packets are sent by a virtual interface.
cospfOriginateLsa	This trap is generated when a new opaque LSA is originated by the router when a topology change has occurred.

CISCO-OSPF-TRAP-MIB Trap Events	Trap Event Description
cospfMaxAgeLsa	The trap is generated in the case of opaque LSAs.
cospfNssaTranslatorStatusChange	The trap is generated if there is a change in the ability of a router to translate OSPF type-7 LSAs into OSPF type-5 LSAs.

For information about how to enable OSPF MIB traps, see the How to Enable OSPF MIB Support of RFC 1850 and Latest Extensions, on page 2807.

Benefits of the OSPF MIB

The OSPF MIBs (OSPF-MIB and OSPF-TRAP-MIB) and Cisco private OSPF MIBs (CISCO-OSPF-MIB and CISCO-OSPF-TRAP-MIB) allow network managers to more effectively monitor the OSPF routing protocol through the addition of new table objects and trap notification objects that previously were not supported by the RFC 1253 OSPF MIB.

New CLI commands have been added to enable SNMP notifications for OSPF MIB support objects, Cisco-specific errors, retransmission and state-change traps. The SNMP notifications are provided for errors and other significant event information for the OSPF network.

How to Enable OSPF MIB Support of RFC 1850 and Latest Extensions

Enabling OSPF MIB Support

Before you begin

Before the OSPF MIB Support of RFC 1850 and Latest Extensions feature can be used, the SNMP server for the router must be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string1* **ro**
4. **snmp-server community** *string2* **rw**
5. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
6. **snmp-server enable traps ospf**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *string1* **ro** **Example:** `Router(config)# snmp-server community public ro`	Enables read access to all objects in the MIB, but does not allow access to the community strings.
Step 4	**snmp-server community** *string2* **rw** **Example:** `Router(config)# snmp-server community private rw`	Enables read and write access to all objects in the MIB, but does not allow access to the community strings.
Step 5	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Router(config)# snmp-server host 172.20.2.162 version 2c public ospf`	Specifies a recipient (target host) for SNMP notification operations. • If no *notification-type* is specified, all enabled notifications (traps or informs) will be sent to the specified host. • If you want to send only the OSPF notifications to the specified host, you can use the optional **ospf**keyword as one of the *notification-types*. (See the example.) Entering the **ospf** keyword enables the ospfSetTrap trap control MIB object.
Step 6	**snmp-server enable traps ospf** **Example:** `Router(config)# snmp-server enable traps ospf`	Enables all SNMP notifications defined in the OSPF MIBs. **Note** This step is required only if you wish to enable all OSPF traps. When you enter the **no snmp-server enable traps ospf** command, all OSPF traps will be disabled.
Step 7	**end** **Example:** `Router(config)# end`	Ends your configuration session and exits global configuration mode.

What to Do Next

If you did not want to enable all OSPF traps, follow the steps in the following section to selectively enable one or more types of OSPF trap:

Enabling Specific OSPF Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps ospf cisco-specific errors** [**config-error**] [**virt-config-error**]
4. **snmp-server enable traps ospf cisco-specific retransmit** [**packets**] [**virt-packets**]
5. **snmp-server enable traps ospf cisco-specific state-change** [**nssa-trans-change**] [**shamlink-state-change**]
6. **snmp-server enable traps ospf cisco-specific lsa** [**lsa-maxage**] [**lsa-originate**]
7. **snmp-server enable traps ospf errors** [**authentication-failure**] [**bad-packet**] [**config-error**] [**virt-authentication-failure**] [**virt-config-error**]
8. **snmp-server enable traps ospf lsa** [**lsa-maxage**] [**lsa-originate**]
9. **snmp-server enable traps ospf rate-limit** *seconds trap-number*
10. **snmp-server enable traps ospf retransmit** [**packets**] [**virt-packets**]
11. **snmp-server enable traps ospf state-change** [**if-state-change**] [**neighbor-state-change**] [**virtif-state-change**] [**virtneighbor-state-change**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps ospf cisco-specific errors** [**config-error**] [**virt-config-error**] **Example:** `Router(config)# snmp-server enable traps ospf cisco-specific errors config-error`	Enables SNMP notifications for Cisco-specific OSPF configuration mismatch errors. • Entering the **snmp-server enable traps ospf cisco-specific errors** command with the optional **virt-config-error** keyword enables only the SNMP notifications for configuration mismatch errors on virtual interfaces.
Step 4	**snmp-server enable traps ospf cisco-specific retransmit** [**packets**] [**virt-packets**]	Enables error traps for Cisco-specific OSPF errors that involve re-sent packets.

	Command or Action	Purpose
	Example: `Router(config)# snmp-server enable traps ospf cisco-specific retransmit packets virt-packets`	• Entering the **snmp-server enable traps ospf cisco-specific retransmit** command with the optional **virt-packets**keyword enables only the SNMP notifications for packets that are re-sent on virtual interfaces.
Step 5	**snmp-server enable traps ospf cisco-specific state-change** [**nssa-trans-change**] [**shamlink-state-change**] **Example:** `Router(config)# snmp-server enable traps ospf cisco-specific state-change`	Enables all error traps for Cisco-specific OSPF transition state changes.
Step 6	**snmp-server enable traps ospf cisco-specific lsa** [**lsa-maxage**] [**lsa-originate**] **Example:** `Router(config)# snmp-server enable traps ospf cisco-specific lsa`	Enables error traps for opaque LSAs.
Step 7	**snmp-server enable traps ospf errors** [**authentication-failure**] [**bad-packet**] [**config-error**] [**virt-authentication-failure**] [**virt-config-error**] **Example:** `Router(config)# snmp-server enable traps ospf errors virt-config-error`	Enables error traps for OSPF configuration errors. • Entering the **snmp-server enable traps ospf errors** command with the optional **virt-config-error**keyword enables only the SNMP notifications for OSPF configuration errors on virtual interfaces.
Step 8	**snmp-server enable traps ospf lsa** [**lsa-maxage**] [**lsa-originate**] **Example:** `Router(config)# snmp-server enable traps ospf lsa`	Enables error traps for OSPF LSA errors.
Step 9	**snmp-server enable traps ospf rate-limit** *seconds trap-number* **Example:** `Router(config)# snmp-server enable traps ospf rate-limit 20 20`	Sets the rate limit for how many SNMP OSPF notifications are sent in each OSPF SNMP notification rate-limit window.
Step 10	**snmp-server enable traps ospf retransmit** [**packets**] [**virt-packets**] **Example:** `Router(config)# snmp-server enable traps ospf retransmit`	Enables SNMP OSPF notifications for re-sent packets.

	Command or Action	Purpose
Step 11	**snmp-server enable traps ospf state-change** [**if-state-change**] [**neighbor-state-change**] [**virtif-state-change**] [**virtneighbor-state-change**] **Example:** `Router(config)# snmp-server enable traps ospf state-change`	Enables SNMP OSPF notifications for OSPF transition state changes.

Verifying OSPF MIB Traps on the Router

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** [*options*] **Example:** `Router# show running-config | include traps`	Displays the contents of the currently running configuration file and includes information about enabled traps. • Verifies which traps are enabled.

Configuration Examples for OSPF MIB Support of RFC 1850 and Latest Extensions

Example Enabling and Verifying OSPF MIB Support Traps

The following example enables all OSPF traps.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# snmp-server enable traps ospf
Router(config)# end
```

The **show running-config** command is entered to verify that the traps are enabled:

```
Router# show running-config | include traps
snmp-server enable traps ospf
```

Where to Go Next

For more information about SNMP and SNMP operations, see the "Configuring SNMP Support" chapter of the Cisco IOS XE Network Management Configuration Guide, *Release 2* .

Additional References

The following sections provide references related to the Area Command in Interface Mode for OSPFv2 feature.

Related Documents

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
OSPF configuration tasks	"Configuring OSPF" module

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2328	*OSPF Version 2*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF MIB Support of RFC 1850 and Latest Extensions

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 268: Feature Information for OSPF MIB Support of RFC 1850 and Latest Extensions

Feature Name	Releases	Feature Information
OSPF MIB Support of RFC 1850 and Latest Extensions	Cisco IOS XE Release 2.1	The OSPF MIB Support of RFC 1850 and Latest Extensions feature introduces the capability for Simple Network Management Protocol (SNMP) monitoring on the Open Shortest Path First (OSPF) routing protocol. Users have an improved ability to constantly monitor the changing state of an OSPF network by use of MIB objects to gather information relating to protocol parameters and trap notification objects that can signal the occurrence of significant network events such as transition state changes. The protocol information collected by the OSPF MIB objects and trap objects can be used to derive statistics that will help monitor and improve overall network performance. The following commands are introduced or modified in the feature documented in this module: • **snmp-server enable traps ospf** • **snmp-server enable traps ospf cisco-specific errors** • **snmp-server enable traps ospf cisco-specific lsa** • **snmp-server enable traps ospf cisco-specific retransmit** • **snmp-server enable traps ospf cisco-specific state-change** • **snmp-server enable traps ospf errors** • **snmp-server enable traps ospf lsa** • **snmp-server enable traps ospf rate-limit** • **snmp-server enable traps ospf retransmit** • **snmp-server enable traps ospf state-change**

Table 269: Feature Information for OSPF MIB Support of RFC 1850 and Latest Extensions

Feature Name	Releases	Feature Information
OSPF MIB Support of RFC 1850 and Latest Extensions	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 227

OSPF Enhanced Traffic Statistics

This document describes new and modified commands that provide enhanced OSPF traffic statistics for OSPFv2 and OSPFv3. The ability to collect and display more detailed traffic statistics increases high availability for the OSPF network by making the troubleshooting process more efficient.

New OSPF traffic statistics are collected and displayed to include the following information:

- OSPF Hello input queue and OSPF process queue status and statistics.
- Global OSPF traffic statistics.
- Per-OSPF-interface traffic statistics.
- Per-OSPF-process traffic statistics.

Prerequisites for OSPF Enhanced Traffic Statistics

OSPFv2 or OSPFv3 must be configured on the router.

Information About OSPF Enhanced Traffic Statistics

The OSPF enhanced traffic statistics are enabled by default and cannot be disabled.

The detailed OSPF traffic statistics are especially beneficial for troubleshooting the following types of OSPF instabilities:

- OSPF process queue status and statistical information can help the network administrator determine if an OSPF process can handle the amount of traffic sent to OSPF.
- OSPF packet header errors and LSA errors statistics keep a record of different errors found in received OSPF packets.

OSPF enhanced traffic control statistics also monitor the amount of traffic control exchanged between OSPF processes--an important consideration in network environments with slow links and frequent topology changes.

How to Display and Clear OSPF Enhanced Traffic Statistics

Displaying and Clearing OSPF Traffic Statistics for OSPFv2

SUMMARY STEPS

1. **enable**
2. **show ip ospf** [*process-id*] **traffic**[*interface-type interface-number*]
3. **clear ip ospf traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip ospf** [*process-id*] **traffic**[*interface-type interface-number*] **Example:** `Router# show ip ospf 10 traffic gigabitethernet 0/0/0`	Displays OSPFv2 traffic statistics.
Step 3	**clear ip ospf traffic** **Example:** `Router# clear ip ospf traffic`	Clears OSPFv2 traffic statistics.

Displaying and Clearing OSPF Traffic Statistics for OSPFv3

SUMMARY STEPS

1. **enable**
2. **show ipv6 ospf** [*process-id*] **traffic**[*interface-type interface-number*]
3. **clear ipv6 ospf traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**show ipv6 ospf** [*process-id*] **traffic**[*interface-type interface-number*] **Example:** `Router# show ipv6 ospf traffic`	Displays OSPFv3 traffic statistics.
Step 3	**clear ipv6 ospf traffic** **Example:** `Router# clear ipv6 ospf traffic`	Clears OSPFv3 traffic statistics.

Configuration Examples for OSPF Enhanced Traffic Statistics

Example Displaying and Clearing Enhanced Traffic Statistics for OSPFv2

The following example shows display output for the **show ip ospf traffic** command for OSPFv2:

```
Router# show ip ospf traffic
OSPF statistics:
 Rcvd: 55 total, 0 checksum errors
        22 hello, 7 database desc, 2 link state req
        6 link state updates, 6 link state acks
  Sent: 68 total
        45 hello, 7 database desc, 2 link state req
        10 link state updates, 4 link state acks
            OSPF Router with ID (10.1.1.1) (Process ID 8)
OSPF queues statistic for process ID 8:
  OSPF Hello queue size 0, no limit, drops 0, max size 0
  OSPF Router queue size 0, limit 200, drops 0, max size 0
Interface statistics:
    Interface GigabitEthernet0/0/1
OSPF packets received/sent
  Type          Packets              Bytes
  RX Invalid    0                    0
  RX Hello      0                    0
  RX DB des     0                    0
  RX LS req     0                    0
  RX LS upd     0                    0
  RX LS ack     0                    0
  RX Total      0                    0
  TX Failed     0                    0
  TX Hello      16                   1216
  TX DB des     0                    0
  TX LS req     0                    0
  TX LS upd     0                    0
  TX LS ack     0                    0
  TX Total      16                   1216
OSPF header errors
```

```
  Length 0, Checksum 0, Version 0, Bad Source 0,
  No Virtual Link 0, Area Mismatch 0, No Sham Link 0,
  Self Originated 0, Duplicate ID 0, Hello 0,
  MTU Mismatch 0, Nbr Ignored 0, LLS 0,
  Authentication 0,
OSPF LSA errors
  Type 0, Length 0, Data 0, Checksum 0,
Summary traffic statistics for process ID 8:
OSPF packets received/sent
  Type          Packets              Bytes
  RX Invalid    0                    0
  RX Hello      0                    0
  RX DB des     0                    0
  RX LS req     0                    0
  RX LS upd     0                    0
  RX LS ack     0                    0
  RX Total      0                    0
  TX Failed     0                    0
  TX Hello      16                   1216
  TX DB des     0                    0
  TX LS req     0                    0
  TX LS upd     0                    0
  TX LS ack     0                    0
  TX Total      16                   1216
OSPF header errors
  Length 0, Checksum 0, Version 0, Bad Source 0,
  No Virtual Link 0, Area Mismatch 0, No Sham Link 0,
  Self Originated 0, Duplicate ID 0, Hello 0,
  MTU Mismatch 0, Nbr Ignored 0, LLS 0,
  Authentication 0,
OSPF LSA errors
  Type 0, Length 0, Data 0, Checksum 0,
           OSPF Router with ID (10.1.1.4) (Process ID 1)
OSPF queues statistic for process ID 1:
  OSPF Hello queue size 0, no limit, drops 0, max size 2
  OSPF Router queue size 0, limit 200, drops 0, max size 2
Interface statistics:
    Interface Serial2/0/0
OSPF packets received/sent
  Type          Packets              Bytes
  RX Invalid    0                    0
  RX Hello      11                   528
  RX DB des     4                    148
  RX LS req     1                    60
  RX LS upd     3                    216
  RX LS ack     2                    128
  RX Total      21                   1080
  TX Failed     0                    0
 TX Hello      14                   1104
  TX DB des     3                    252
  TX LS req     1                    56
  TX LS upd     3                    392
  TX LS ack     2                    128
  TX Total      23                   1932
OSPF header errors
  Length 0, Checksum 0, Version 0, Bad Source 0,
  No Virtual Link 0, Area Mismatch 0, No Sham Link 0,
  Self Originated 0, Duplicate ID 0, Hello 0,
  MTU Mismatch 0, Nbr Ignored 0, LLS 0,
  Authentication 0,
OSPF LSA errors
  Type 0, Length 0, Data 0, Checksum 0,
    Interface GigabitEthernet0/0/0
OSPF packets received/sent
```

```
  Type          Packets              Bytes
  RX Invalid    0                    0
  RX Hello      13                   620
  RX DB des     3                    116
  RX LS req     1                    36
  RX LS upd     3                    228
  RX LS ack     4                    216
  RX Total      24                   1216
  TX Failed     0                    0
  TX Hello      17                   1344
  TX DB des     4                    276
  TX LS req     1                    56
  TX LS upd     7                    656
  TX LS ack     2                    128
  TX Total      31                   2460
OSPF header errors
  Length 0, Checksum 0, Version 0, Bad Source 13,
  No Virtual Link 0, Area Mismatch 0, No Sham Link 0,
  Self Originated 0, Duplicate ID 0, Hello 0,
  MTU Mismatch 0, Nbr Ignored 0, LLS 0,
  Authentication 0,
OSPF LSA errors
  Type 0, Length 0, Data 0, Checksum 0,

Summary traffic statistics for process ID 1:
OSPF packets received/sent
  Type          Packets              Bytes
  RX Invalid    0                    0
  RX Hello      24                   1148
  RX DB des     7                    264
  RX LS req     2                    96
  RX LS upd     6                    444
  RX LS ack     6                    344
  RX Total      45                   2296
  TX Failed     0                    0
  TX Hello      31                   2448
  TX DB des     7                    528
  TX LS req     2                    112
  TX LS upd     10                   1048
  TX LS ack     4                    256
  TX Total      54                   4392
OSPF header errors
  Length 0, Checksum 0, Version 0, Bad Source 13,
  No Virtual Link 0, Area Mismatch 0, No Sham Link 0,
  Self Originated 0, Duplicate ID 0, Hello 0,
  MTU Mismatch 0, Nbr Ignored 0, LLS 0,
  Authentication 0,
OSPF LSA errors
  Type 0, Length 0, Data 0, Checksum 0,
```

The network administrator can issue the **clear ip ospf traffic** command to reset all counters and restart all statistics collections:

```
Router# clear ip ospf traffic
```

Example Displaying and Clearing Enhanced Traffic Statistics for OSPFv3

The following example shows display output for the **show ipv6 ospf traffic** command for OSPFv3:

```
Router# show ipv6 ospf traffic
```

```
OSPFv3 statistics:
  Rcvd: 32 total, 0 checksum errors
        10 hello, 7 database desc, 2 link state req
        9 link state updates, 4 link state acks
        0 LSA ignored
  Sent: 45 total, 0 failed
        17 hello, 12 database desc, 2 link state req
        8 link state updates, 6 link state acks
            OSPFv3 Router with ID (10.1.1.4) (Process ID 6)
OSPFv3 queues statistic for process ID 6
  Hello queue size 0, no limit, max size 2
  Router queue size 0, limit 200, drops 0, max size 2
Interface statistics:
    Interface Serial2/0/0
OSPFv3 packets received/sent
  Type          Packets              Bytes
  RX Invalid    0                    0
  RX Hello      5                    196
  RX DB des     4                    172
  RX LS req     1                    52
  RX LS upd     4                    320
  RX LS ack     2                    112
  RX Total      16                   852
  TX Failed     0                    0
  TX Hello      8                    304
  TX DB des     3                    144
  TX LS req     1                    52
  TX LS upd     3                    252
  TX LS ack     3                    148
  TX Total      18                   900
OSPFv3 header errors
  Length 0, Checksum 0, Version 0, No Virtual Link 0,
  Area Mismatch 0, Self Originated 0, Duplicate ID 0,
  Instance ID 0, Hello 0, MTU Mismatch 0,
  Nbr Ignored 0, Authentication 0,
OSPFv3 LSA errors
  Type 0, Length 0, Data 0, Checksum 0,
Interface GigabitEthernet0/0/0
OSPFv3 packets received/sent
  Type          Packets              Bytes
  RX Invalid    0                    0
  RX Hello      6                    240
  RX DB des     3                    144
  RX LS req     1                    52
  RX LS upd     5                    372
  RX LS ack     2                    152
  RX Total      17                   960
  TX Failed     0                    0
  TX Hello      11                   420
  TX DB des     9                    312
  TX LS req     1                    52
  TX LS upd     5                    376
  TX LS ack     3                    148
  TX Total      29                   1308
OSPFv3 header errors
  Length 0, Checksum 0, Version 0, No Virtual Link 0,
  Area Mismatch 0, Self Originated 0, Duplicate ID 0,
  Instance ID 0, Hello 0, MTU Mismatch 0,
  Nbr Ignored 0, Authentication 0,
OSPFv3 LSA errors
  Type 0, Length 0, Data 0, Checksum 0,
Summary traffic statistics for process ID 6:
OSPFv3 packets received/sent
  Type          Packets              Bytes
```

```
  RX Invalid    0                   0
  RX Hello      11                  436
  RX DB des     7                   316
  RX LS req     2                   104
  RX LS upd     9                   692
  RX LS ack     4                   264
  RX Total      33                  1812
  TX Failed     0                   0
  TX Hello      19                  724
  TX DB des     12                  456
  TX LS req     2                   104
  TX LS upd     8                   628
  TX LS ack     6                   296
  TX Total      47                  2208
OSPFv3 header errors
  Length 0, Checksum 0, Version 0, No Virtual Link 0,
  Area Mismatch 0, Self Originated 0, Duplicate ID 0,
  Instance ID 0, Hello 0, MTU Mismatch 0,
  Nbr Ignored 0, Authentication 0,
OSPFv3 LSA errors
  Type 0, Length 0, Data 0, Checksum 0,
```

The network administrator can issue the **clear ipv6 ospf traffic** command to reset all counters and restart all statistics collections:

```
Router# clear ipv6 ospf traffic
```

Additional References

The following sections provide references related to the OSPF Sham-Link MIB Support feature.

Related Documents

Related Topic	Document Title
Configuring OSPF sham-links	OSPF Sham-Link Support for MPLS VPN
SNMP configuration	*Cisco IOS Network Management Configuration Guide.*
SNMP commands	*Cisco IOS Network Management Command Reference.*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
• CISCO-OSPF-MIB • CISCO-OSPF-TRAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for OSPF Enhanced Traffic Statistics

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 228

TTL Security Support for OSPFv3 on IPv6

The Time To Live (TTL) Security Support for Open Shortest Path First version 3 (OSPFv3) on IPv6 feature increases protection against OSPFv3 denial of service attacks.

Restrictions for TTL Security Support for OSPFv3 on IPv6

- OSPFv3 TTL security can be configured for virtual and sham links only.
- OSPFv3 TTL security must be configured in IPv6 address family configuration mode (config-router-af). To enter IPv6 address family configuration mode you use the **address-family ipv6** command.
- Sham links must not be configured on the default Virtual Routing and Forwarding (VRF).

Prerequisites for TTL Security Support for OSPFv3 on IPv6

The TTL Security Support for OSPFv3 on IPv6 feature is available only on platforms with OSPFv3 routing capabilities.

Information About TTL Security Support for OSPFv3 on IPv6

OSPFv3 TTL Security Support for Virtual and Sham Links

In OSPFv3, all areas must be connected to a backbone area. If there is a break in backbone continuity, or the backbone is purposefully partitioned, you can establish a virtual link. The virtual link must be configured in the two devices you want to use to connect the partitioned backbone. The configuration information in each

device consists of the other virtual endpoint (the other Area Border Router [ABR]) and the nonbackbone area that the two devices have in common (called the transit area.) Note that virtual links cannot be configured through stub areas. Sham links are similar to virtual links in many ways, but sham links are used in Layer 3 Multiprotocol Label Switching (MPLS) VPN networks to connect provider edge (PE) routers across the MPLS backbone.

Note Multihop adjacencies such as virtual links and sham links use global IPv6 addresses that require you to configure TTL security to control the number of hops that a packet can travel.

If TTL security is enabled, OSPFv3 sends outgoing packets with an IP header TTL value of 255 and discards incoming packets that have TTL values less than the configurable threshold. Because each device that forwards an IP packet decreases the TTL value, packets received via a direct (one-hop) connection will have a value of 255. Packets that cross two hops will have a value of 254, and so on. The receive threshold is configured in terms of the maximum number of hops that a packet may have traveled. The value for this *hop-count* argument is a number from 1 to 254, with a default of 1.

To establish a virtual link or a sham link, use the **area virtual-link** or **area sham-link cost** commands respectively. To configure TTL security on a virtual link or a sham link, configure the **ttl-security** keyword and the *hop-count* argument in either command. Note that the *hop-count* argument value is mandatory in this case.

Note OSPFv3 TTL Security can be configured for virtual and sham links only, and must be configured in address family configuration (config-router-af) mode for IPv6 address families.

How to Configure TTL Security Support for OSPFv3 on IPv6

Configuring TTL Security Support on Virtual Links for OSPFv3 on IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **address-family ipv6 unicast vrf** *vrf-name*
5. **area** *area-ID* **virtual-link** *router-id* **ttl-security hops** *hop-count*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables router configuration mode for the IPv4 or IPv6 address family.
Step 4	**address-family ipv6 unicast vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv6 unicast vrf vrf1`	Enters address family configuration mode for OSPFv3, specifies IPv6 unicast address prefixes, and specifies the name of the VRF instance to associate with subsequent address family configuration mode commands.
Step 5	**area** *area-ID* **virtual-link** *router-id* **ttl-security hops** *hop-count* **Example:** `Device(config-router-af)# area 1 virtual-link 10.1.1.1 ttl-security hops 10`	Defines an OSPFv3 virtual link and configures TTL security on the virtual link.
Step 6	**end** **Example:** `Device(config-router-af)# end`	(Optional) Returns to privileged EXEC mode.

Configuring TTL Security Support on Sham Links for OSPFv3 on IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. **address-family ipv6 unicast vrf** *vrf-name*
5. **area** *area-id* **sham-link** *source-address destination-address* **ttl-security hops** *hop-count*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	**address-family ipv6 unicast vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv6 unicast vrf vrf1`	Enters address family configuration mode for OSPFv3, specifies IPv6 unicast address prefixes, and specifies the name of the VRF instance to associate with subsequent address family configuration mode commands.
Step 5	**area** *area-id* **sham-link** *source-address destination-address* **ttl-security hops** *hop-count* **Example:** `Device(config-router-af)# area 1 sham-link 2001:DB8:1::1 2001:DB8:0:A222::2 ttl-security hops 10`	Defines an OSPFv3 sham link and configures TTL security on the sham link.
Step 6	**end** **Example:** `Device(config-router-af)# end`	(Optional) Returns to privileged EXEC mode.

Configuration Examples for TTL Security Support for OSPFv3 on IPv6

Example: TTL Security Support on Virtual Links for OSPFv3 on IPv6

The following example shows how to configure TTL virtual link security:

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# address-family ipv6 unicast vrf vrf1
Device(config-router-af)# area 1 virtual-link 10.1.1.1 ttl-security hops 10
Device(config-router-af)# end
```

```
Device# show ospfv3 virtual-links
OSPFv3 1 address-family ipv6 (router-id 10.1.1.7)
Virtual Link OSPFv3_VL0 to router 10.1.1.2 is down
  Interface ID 23, IPv6 address ::
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 1, Cost of using 65535
  Transmit Delay is 1 sec, State DOWN,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Strict TTL checking enabled, up to 10 hops allowed
```

Example: TTL Security Support on Sham Links for OSPFv3 on IPv6

The following example shows how to configure TTL sham link security:

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# address-family ipv6 unicast vrf vrf1
Device(config-router-af)# area 1 sham-link 2001:DB8:1::1 2001:DB8:0:A222::2 ttl-security
hops 10
Device(config-router-af)# end
Device#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
IPv6 routing: OSPFv3	"IPv6 Routing: OSPFv3" module

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for TTL Security Support for OSPFv3 on IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 270: TTL Security Support for OSPFv3 on IPv6

Feature Name	Software Releases	Feature Information
TTL Security Support for OSPFv3 on IPv6	Cisco IOS XE Release 3.7S	The TTL Security Support for OSPFv3 on IPv6 feature increases protection against OSPFv3 denial of service attacks. The following commands were introduced or modified by this feature: **area sham-link**, **area virtual-link**.

Table 271: TTL Security Support for OSPFv3 on IPv6

Feature Name	Software Releases	Feature Information
TTL Security Support for OSPFv3 on IPv6	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 229

Configuring OSPF TTL Security Check and OSPF Graceful Shutdown

This module describes configuration tasks to configure various options involving Open Shortest Path First (OSPF). This module contains tasks that use commands to configure a lightweight security mechanism to protect OSPF sessions from CPU-utilization-based attacks and to configure a router to shut down a protocol temporarily without losing the protocol configuration.

Information About OSPF TTL Security Check and OSPF Graceful Shutdown

TTL Security Check for OSPF

When the TTL Security Check feature is enabled, OSPF sends outgoing packets with an IP header Time to Live (TTL) value of 255 and discards incoming packets that have TTL values less than a configurable threshold. Since each device that forwards an IP packet decrements the TTL, packets received via a direct (one-hop) connection will have a value of 255. Packets that cross two hops will have a value of 254, and so on. The receive threshold is configured in terms of the maximum number of hops that a packet may have traveled. The value for this *hop-count* argument is a number from 1 to 254, with a default of 1.

The TTL Security Check feature may be configured under the OSPF router submode, in which case it applies to all the interfaces on which OSPF runs, or it may be configured on a per-interface basis.

Transitioning Existing Networks to Use TTL Security Check

If you currently have OSPF running in your network and want to implement TTL security on an interface-by-interface basis without any network interruptions, use the **ip ospf ttl-security** command and set the hop-count argument to 254. This setting causes outgoing packets to be sent with a TTL value of 255, but

allows any value for input packets. Later, once the device at the other end of the link has had TTL security enabled you can start enforcing the hop limit for the incoming packets by using the same **ip ospf ttl-security** command with no hop count specified. This process ensures that OSPF packets will not be dropped because of a temporary mismatch in TTL security.

TTL Security Check for OSPF Virtual and Sham Links

In OSPF, all areas must be connected to a backbone area. If there is a break in backbone continuity, or the backbone is purposefully partitioned, you can establish a *virtual link*. The virtual link must be configured in both devices. The configuration information in each device consists of the other virtual endpoint (the other area border router [ABR]) and the nonbackbone area that the two devices have in common (called the *transit area*.) Note that virtual links cannot be configured through stub areas. Sham links are similar to virtual links in many ways, but sham links are used in Layer 3 Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) networks to connect Provider Edge (PE) routers across the MPLS backbone.

To establish a virtual link or a sham link, use the **area virtual-link** or **area sham-link cost** commands, respectively, in router configuration mode. To configure the TTL Security Check feature on a virtual link or a sham link, configure the **ttl-security** keyword and the *hop-count* argument in either command. Note that the *hop-count* argument value is mandatory in this case.

Benefits of the OSPF Support for TTL Security Check

The OSPF Support for TTL Security Check feature provides an effective and easy-to-deploy solution to protect OSPF neighbor sessions from CPU utilization-based attacks. When this feature is enabled, a host cannot attack an OSPF session if the host is not a member of the local or remote OSPF network, or if the host is not directly connected to a network segment between the local and remote OSPF networks. This solution greatly reduces the effectiveness of Denial of Service (DoS) attacks against an OSPF autonomous system.

OSPF Graceful Shutdown

The OSPF Graceful Shutdown feature provides the ability to temporarily shut down the OSPF protocol in the least disruptive manner and notify its neighbors that it is going away. All traffic that has another path through the network will be directed to that alternate path. A graceful shutdown of the OSPF protocol can be initiated using the **shutdown** command in router configuration mode.

This feature also provides the ability to shut down OSPF on a specific interface. In this case, OSPF will not advertise the interface or form adjacencies over it; however, all of the OSPF interface configuration will be retained. To initiate a graceful shutdown of an interface, use the **ip ospf shutdown** command in interface configuration mode.

How to Configure OSPF TTL Security Check and OSPF Graceful Shutdown

Configuring TTL Security Check on All OSPF Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **ttl-security all-interfaces** [**hops** *hop-count*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing, which places the device in router configuration mode.
Step 4	**ttl-security all-interfaces** [**hops** *hop-count*] **Example:** `Device(config-router)# ttl-security all-interfaces`	Configures TTL security check on all OSPF interfaces. **Note** This configuration step applies only to normal OSPF interfaces. This step does not apply to virtual links or sham links that require TTL security protection. Virtual links and sham links must be configured independently.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.

Configuring TTL Security Check on a Per-Interface Basis

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf ttl-security** [**hops** *hop-count* | **disable**]
5. **end**
6. **show ip ospf** [*process-id*] **interface** [*interface type interface-number*] [**brief**] [**multicast**] [**topology** *topology-name* | **base**}]
7. **show ip ospf neighbor** *interface-type interface-number* [*neighbor-id*][**detail**]
8. **show ip ospf** [*process-id*] **traffic** [*interface-type interface-number*]
9. **debug ip ospf adj**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip ospf ttl-security** [**hops** *hop-count* | **disable**] **Example:** `Device(config-if)# ip ospf ttl-security`	Configures TTL security check feature on a specific interface. • The *hop-count*argument range is from 1 to 254. • The **disable** keyword can be used to disable TTL security on an interface. It is useful only if the **ttl-security all-interfaces** comand initially enabled TTL security on all OSPF interfaces, in which case **disable** can be used as an override or to turn off TTL security on a specific interface. • In the example, TTL security is being disabled on GigabitEthernet interface 0/0/0.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip ospf** [*process-id*] **interface** [*interface type interface-number*] [**brief**] [**multicast**] [**topology** *topology-name* | **base**}] **Example:** `Device# show ip ospf interface gigabitethernet 0/0/0`	(Optional) Displays OSPF-related interface information.
Step 7	**show ip ospf neighbor** *interface-type interface-number* [*neighbor-id*][**detail**] **Example:** `Device# show ip ospf neighbor 10.199.199.137`	(Optional) Displays OSPF neighbor information on a per-interface basis. • If one side of the connection has TTL security enabled, the other side shows the neighbor in the INIT state.
Step 8	**show ip ospf** [*process-id*] **traffic** [*interface-type interface-number*] **Example:** `Device# show ip ospf traffic`	(Optional) Displays OSPF traffic statistics. • The number of times a TTL security check failed is included in the output.
Step 9	**debug ip ospf adj** **Example:** `Device# debug ip ospf adj`	(Optional) Initiates debugging of OSPF adjacency events. • Information about dropped packets, including interface type and number, neighbor IP address, and TTL value, is included in the command output.

Configuring OSPF Graceful Shutdown on a Per-Interface Basis

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf shutdown**
5. **end**
6. **show ip ospf [** *process-id* **] interface [** *interface type interface-number* **] [brief** *]* [**multicast**] [**topology** *topology-name* | **base**}]
7. **show ip ospf [** *process-id* **]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/0`	Configures an interface type and number and enters interface configuration mode.
Step 4	**ip ospf shutdown** **Example:** `Device(config-if)# ip ospf shutdown`	Initiates an OSPF protocol graceful shutdown at the interface level. • When the **ip ospf shutdown** interface command is entered, the interface on which it is configured sends a link-state update advising its neighbors that is going down, which allows those neighbors to begin routing OSPF traffic around this router.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip ospf [** *process-id* **] interface [** *interface type interface-number* **]** [**brief**] [**multicast**] [**topology** *topology-name* | **base**}] **Example:** `Device# show ip ospf interface GigabitEthernet 0/1/0`	(Optional) Displays OSPF-related interface information.
Step 7	**show ip ospf [** *process-id* **]** **Example:** `Device# show ip ospf`	(Optional) Displays general information about OSPF routing processes.

Configuration Examples for OSPF TTL Security Check and OSPF Graceful Shutdown

Example: Transitioning an Existing Network to Use TTL Security Check

The following example shows how to enable TTL security in an existing OSPF network on a per-interface basis.

Configuring TTL security in an existing network is a three-step process:

1. Configure TTL security with a hop count of 254 on the OSPF interface on the sending side device.
2. Configure TTL security with no hop count on the OSPF interface on the receiving side device.
3. Reconfigure the sending side OSPF interface with no hop count.

```
configure terminal
 ! Configure the following command on the sending side router.
 interface gigabitethernet 0/1/0
  ip ospf ttl-security hops 254
 ! Configure the next command on the receiving side router.
 interface gigabitethernet 0/1/0
  ip ospf ttl-security
 ! Reconfigure the sending side with no hop count.
  ip ospf ttl-security
  end
```

Additional References

The following sections provide references related to the OSPF TTL Security Check and OSPF Graceful Shutdown features.

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported and support for existing MIBs has not been modified.	http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring OSPF TTL Security Check and OSPF Graceful Shutdown

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 230

OSPF Sham-Link MIB Support

This feature introduces MIB support for the OSPF Sham-Link feature through the addition of new tables and trap MIB objects to the Cisco OSPF MIB (CISCO-OSPF-MIB) and the Cisco OSPF Trap MIB (CISCO-OSPF-TRAP-MIB). New commands have been added to enable Simple Network Management Protocol (SNMP) notifications for the Open Shortest Path First (OSPF) sham-link trap objects. Notifications are provided for errors, state changes, and retransmissions across a sham-link interface.

Prerequisites for OSPF Sham-Link MIB Support

- It is presumed that you already have configured an OSPF sham-link.
- SNMP must be enabled on the router before notifications (traps) can be configured or before SNMP GET operations can be performed.

Restrictions for OSPF Sham-Link MIB Support

All enhancements that are introduced by this feature are provided only by the Cisco private MIBs CISCO-OSPF-MIB and CISCO-OSPF-TRAP-MIB.

Information About OSPF Sham-Link MIB Support

OSPF Sham-Links in PE-PE Router Connections

In a Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) configuration, a virtual connection called a sham-link can be configured to interconnect two VPN sites that want to be in the same OSPF area. The sham-link is configured on top of the MPLS VPN tunnel that connects two provider edge (PE) routers. The OSPF packets are propagated over the sham-link. For more information on configuring sham-links, see the " OSPF Sham-Link Support for MPLS VPN" chapter.

Cisco OSPF MIB and Cisco OSPF Trap MIB Enhancements

The OSPF Sham-Link MIB Support feature introduces MIB support for OSPF sham-links through the addition of new tables and trap MIB objects to the Cisco OSPF MIB (CISCO-OSPF-MIB) and the Cisco OSPF Trap MIB (CISCO-OSPF-TRAP-MIB). New command-line interface (CLI) commands have been added to enable SNMP notifications for the OSPF sham-link trap objects. Notifications are provided for errors, state changes, and retransmissions across a sham-link interface.

OSPF Sham-Link Configuration Support

The cospfShamLinksTable table object stores information about the sham-links that have been configured for the OSPF area. The cospfShamLinksTable allows access to the following MIB objects:

- cospfShamLinksAreaId
- cospfShamLinksLocalIpAddrType
- cospfShamLinksLocalIpAddr
- cospfShamLinksRemoteIpAddrType
- cospfShamLinksRemoteIpAddr
- cospfShamLinksRetransInterval
- cospfShamLinksHelloInterval
- cospfShamLinksRtrDeadInterval
- cospfShamLinksState
- cospfShamLinksEvents
- cospfShamLinksMetric

OSPF Sham-Link Neighbor Support

The cospfShamLinkNbrTable table object describes all OSPF sham-link neighbor entries. The cospfShamLinkNbrTable allows access to the following MIB objects:

- cospfShamLinkNbrArea
- cospfShamLinkNbrIpAddrType

- cospfShamLinkNbrIpAddr
- cospfShamLinkNbrRtrId
- cospfShamLinkNbrOptions
- cospfShamLinkNbrState
- cospfShamLinkNbrEvents
- cospfShamLinkNbrLsRetransQLen
- cospfShamLinkNbrHelloSuppressed

OSPF Sham-Link Interface Transition State Change Support

The cospfShamLinksStateChange trap object is used to notify the network manager of a transition state change for the OSPF sham-link interface. The cospfShamLinksStateChange trap objects contains the following MIB objects:

- ospfRouterId
- cospfShamLinksAreaId
- cospfShamLinksLocalIpAddrType
- cospfShamLinksLocalIpAddr
- cospfShamLinksRemoteIpAddrType
- cospfShamLinksRemoteIpAddr
- cospfShamLinksState

OSPF Sham-Link Neighbor Transition State Change Support

The cospfShamLinkNbrStateChange trap object is used to notify the network manager of a transition state change for the OSPF sham-link neighbors. The cospfShamLinkNbrStateChange trap object contains the following MIB objects:

- ospfRouterId
- cospfShamLinkNbrArea
- cospfShamLinksLocalIpAddrType
- cospfShamLinksLocalIpAddr
- cospfShamLinkNbrIpAddrType
- cospfShamLinkNbrIpAddr
- cospfShamLinkNbrRtrId
- cospfShamLinkNbrState

Sham-Link Errors

Trap notifications are provided for OSPF sham-link configuration, authentication, and bad packet errors. These errors include the following trap objects:

- cospfShamLinkConfigError
- cospfShamLinkAuthFailure
- cospfShamLinkRxBadPacket

How to Configure OSPF Sham-Link MIB Support

Configuring the Router to Enable Sending of SNMP Notifications

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
5. **snmp-server enable traps ospf**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration to determine if an SNMP agent is already running. • If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as needed.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** |	Specifies a recipient (target host) for SNMP notification operations.

	Command or Action	Purpose
	priv]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Router(config)# snmp-server host 172.20.2.162 version 2c public ospf`	• If no *notification-type* is specified, all enabled notifications (traps or informs) will be sent to the specified host. • If you want to send only the OSPF notifications to the specified host, you can use the optional **ospf**keyword as one of the *notification-types*. (See the example.)
Step 5	**snmp-server enable traps ospf** **Example:** `Router(config)# snmp-server enable traps ospf`	Enables all SNMP notifications defined in the OSPF MIBs. **Note** This step is required only if you want to enable all OSPF traps, including the traps for OSPF sham-links. When you enter the **no snmp-server enable traps ospf** command, all OSPF traps, including the OSPF sham-link trap, will be disabled.
Step 6	**end** **Example:** `Router(config)# end`	Ends your configuration session and exits global configuration mode.

Enabling Sending of OSPF Sham-Link Error Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps ospf cisco-specific errors config-error**
4. **snmp-server enable traps ospf cisco-specific errors shamlink** [**authentication** [**bad-packet** [**config**] | **config** [**bad-packet**]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**snmp-server enable traps ospf cisco-specific errors config-error** **Example:** `Router(config)# snmp-server enable traps ospf cisco-specific errors config-error`	Enables error traps for OSPF nonvirtual interface mismatch errors. **Note** You must enter the **snmp-server enable traps ospf cisco-specific errors config-error** command before you enter the **snmp-server enable traps ospf cisco-specific errors shamlink** command, in order for both traps to be generated at the same place and maintain consistency with a similar case for configuration errors across virtual links. If you try to enable the cospfShamLinkConfigError trap before configuring the cospfospfConfigError trap you will receive an error message stating you must first configure the cospfConfigError trap.
Step 4	**snmp-server enable traps ospf cisco-specific errors shamlink** [**authentication** [**bad-packet** [**config**] | **config** [**bad-packet**]] **Example:** `Router(config)# snmp-server enable traps ospf cisco-specific errors shamlink`	Enables error traps for OSPF sham-link errors. • The **authentication** keyword enables SNMP notifications only for authentication failures on OSPF sham-link interfaces. • The **bad-packet** keyword enables SNMP notifications only for packet parsing failures on OSPF sham-link interfaces. • The **config** keyword enables SNMP notifications only for configuration mismatch errors on OSPF sham-link interfaces.
Step 5	**end** **Example:** `Router(config)# end`	Ends your configuration session and exits global configuration mode.

Enabling OSPF Sham-Link Retransmissions Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps ospf cisco-specific retransmit** [**packets** [**shamlink** | **virt-packets**] | **shamlink** [**packets** | **virt-packets**] | **virt-packets** [**shamlink**]]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps ospf cisco-specific retransmit** [**packets** [**shamlink** | **virt-packets**] | **shamlink** [**packets** | **virt-packets**] | **virt-packets** [**shamlink**]] **Example:** `Router(config)# snmp-server enable traps ospf cisco-specific retransmit shamlink`	Enables error traps for OSPF sham-link retransmission errors.
Step 4	**end** **Example:** `Router(config)# end`	Ends your configuration session and exits global configuration mode.

Enabling OSPF Sham-Link State Change Traps

Note The replaced cospfShamLinkChange trap can still be enabled, but not when you want to enable the new cospfShamLinksStateChange trap.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps ospf cisco-specific state-change** [**nssa-trans-change** | **shamlink** [**interface** | **interface-old** | **neighbor**]]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps ospf cisco-specific state-change** [**nssa-trans-change** | **shamlink** [**interface** | **interface-old** | **neighbor**]] **Example:** `Router(config)# snmp-server enable traps ospf cisco-specific state-change`	Enables all Cisco-specific OSPF state change traps including the cospfShamLinksStateChange and cospfShamLinkNbrStateChange traps. • The **neighbor** keyword enables the OSPF sham-link neighbor state change traps. • The **interface** keyword enables the OSPF sham-link interface state change traps. • The **interface-old** keyword enables the original OSPF sham-link interface state change trap that is replaced by the cospfShamLinksStateChange and cospfShamLinkNbrStateChange traps. **Note** You cannot enter both the **interface** and **interface-old** keywords because you cannot enable both the new and replaced sham-link interface transition state change traps. You can configure only one of the two traps, but not both.
Step 4	**end** **Example:** `Router(config)# end`	Ends your configuration session and exits global configuration mode.

Verifying OSPF Sham-Link MIB Traps on the Router

SUMMARY STEPS

1. **enable**
2. **show running-config** | **include traps**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show running-config** | **include traps** **Example:** `Router# show running-config | include traps`	Displays the contents of the currently running configuration file and includes information about enabled traps. • Verifies if the trap is enabled.

Configuration Examples for OSPF Sham-Link MIB Support

Example Enabling and Verifying OSPF Sham-Link Error Traps

The following example enables all Cisco-specific OSPF sham-link error traps. Note that the first attempt to enter the **snmp-server enable traps ospf cisco-specific errors shamlink** command results in an error message that the **snmp-server enable traps ospf cisco-specific errors config-error** command must be entered first:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# snmp-server enable traps ospf cisco-specific errors shamlink

% Sham-link config error trap not enabled.
% Configure "cisco-specific errors config-error" first.
% This requirement allows both traps to be sent.
Router(config)# snmp-server enable traps ospf cisco-specific errors config-error
Router(config)# snmp-server enable traps ospf cisco-specific errors shamlink
Router(config)# end
```

The **show running-config** command is entered to verify that the traps are enabled:

```
Router# show running-config | include traps
snmp-server enable traps ospf cisco-specific errors config-error
snmp-server enable traps ospf cisco-specific errors shamlink
```

At the time of disabling the traps, if the **no snmp-server enable traps ospf cisco-specific errors config-error** command is entered before the **snmp-server enable traps ospf cisco-specific errors shamlink** command, a message will be displayed to indicate that the sham-link configuration errors traps have also been disabled:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# no snmp-server enable traps ospf cisco-specific errors config-error
! This command also disables the previously-enabled shamlink configuration error traps.
Router(config)# end
```

Example Enabling and Verifying OSPF State Change Traps

The following example enables all Cisco-specific OSPF state change traps including the cospfShamLinksStateChange and cospfShamLinkNbrStateChange traps:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# snmp-server enable traps ospf cisco-specific state-change shamlink
```

The **show running-config** command is entered to verify that the traps are enabled:

```
Router# show running-config | include traps
snmp-server enable traps ospf cisco-specific state-change shamlink interface
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
```

Note that the **snmp-server enable traps ospf cisco-specific state-change shamlink** command enables the sham-link interface state change for the cospfShamLinksStateChange trap.

To enable the original cospfShamLinkStateChange trap, you must first disable the cospfShamLinksStateChange trap. An attempt to enter the **snmp-server enable traps ospf cisco-specific state-change shamlink interface-old** command results in the following error message:

```
Router(config)# snmp-server enable traps ospf cisco-specific state-change shamlink
interface-old
% Cannot enable both sham-link state-change interface traps.
% Deprecated sham link interface trap not enabled.
Router(config)# no snmp-server enable traps ospf cisco-specific state-change shamlink
interface
Router(config)# snmp-server enable traps ospf cisco-specific state-change shamlink
interface-old
```

Example Enabling and Verifying OSPF Sham-Link Retransmissions Traps

The following example enables all OSPF sham-link retransmissions traps:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# snmp-server enable traps ospf cisco-specific retransmit shamlink
Router(config)# end
```

The **show running-config** command is entered to verify that the traps are enabled:

```
Router# show running-config | include traps
snmp-server enable traps ospf cisco-specific retransmit shamlink
```

Where to Go Next

For more information about SNMP and SNMP operations, see the "Configuring SNMP Support" part of the *Cisco IOS XE Network Management Configuration Guide, Release 2* .

Additional References

The following sections provide references related to the OSPF Sham-Link MIB Support feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF sham-links	OSPF Sham-Link Support for MPLS VPN
SNMP configuration	"Configuring SNMP Support"
SNMP commands	*Cisco IOS Network Management Command Reference*
Configuring OSPF	Configuring OSPF
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
• CISCO-OSPF-MIB • CISCO-OSPF-TRAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Sham-Link MIB Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 272: Feature Information for OSPF Sham-Link MIB Support

Feature Name	Releases	Feature Information
OSPF Sham-Link MIB Support	Cisco IOS XE Release 2.1 Cisco IOS XE Release 2.6	This feature introduces MIB support for the OSPF Sham-Link feature through the addition of new tables and trap MIB objects to the Cisco OSPF MIB (CISCO-OSPF-MIB) and to the Cisco OSPF Trap MIB (CISCO-OSPF-TRAP-MIB). New commands have been added to enable Simple Network Management Protocol (SNMP) notifications for the Open Shortest Path First (OSPF) sham-link trap objects. Notifications are provided for errors, state changes, and retransmissions across a sham-link interface. The following commands are introduced or modified in the feature documented in this module: • **snmp-server enable traps ospf cisco-specific errors config-error** • **snmp-server enable traps ospf cisco-specific errors shamlink** • **snmp-server enable traps ospf cisco-specific retransmit** • **snmp-server enable traps ospf cisco-specific state-change.**

Table 273: Feature Information for OSPF Sham-Link MIB Support

Feature Name	Releases	Feature Information
OSPF Sham-Link MIB Support	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 231

OSPF SNMP ifIndex Value for Interface ID in Data Fields

This feature allows you to configure the interface ID value Open Shortest Path First version 2 (OSPFv2) and Open Shortest Path First version 3 (OSPFv3) data fields. You can choose to use either the current interface number or the Simple Network Management Protocol (SNMP) MIB-II interface index (ifIndex) value for the interface ID. The advantage to using the SNMP MIB-II ifIndex value is that this number corresponds to the number that the user will see reported by SNMP.

Prerequisites for SNMP ifIndex Value for Interface ID in Data Fields

Before you can use the SNMP ifIndex value for interface identification, OSPF must be configured on the router.

Information About SNMP ifIndex Value for Interface ID in Data Fields

Benefits of Choosing to Identify Interfaces by the SNMP MIB-II ifIndex Value

If you use SNMP for your OSPF network, configuring the OSPF: SNMP ifIndex Value for Interface ID in OSPFv2 and OSPFv3 Data Fields feature can be beneficial for the following reasons:

- Using the SNMP MIB-II ifIndex identification numbers to identify OSPF interfaces makes it easier for network administrators to identify interfaces because the numbers will correspond to the numbers that they will see reported by SNMP.

- In the link-state advertisements (LSAs), the value used in fields that have the interface ID will be the same as the value that is reported by SNMP.
- In the output from the **show ipv6 ospf interface** command, the interface ID number will have the same value that is reported by SNMP.
- Using the SNMP MIB-II IfIndex is also suggested, but not required, by the OSPF RFC 2328 for OSPFv2 and the RFC 2740 for OSPFv3.

How OSPFv2 and OSPFv3 Use the SNMP MIB-II ifIndex Value

The user chooses for OSPF interfaces to use the SNMP MIB-II ifIndex number by entering the **interface-id snmp-if-index** command for a specific OSPF process. If an interface under the specific OSPF process does not have an SNMP ifIndex number, OSPF will not be enabled on that interface.

For OSPFv2, the ifIndex number is used for the Link Data field in the Router LSA for unnumbered point-to-point interfaces and sham links. When the **interface-id snmp-if-index** command is entered, the affected LSAs will immediately be reoriginated.

For OSPFv3, the ifIndex number is used for the interface ID in router LSAs, as the LSID in Network and Link LSAs, and also as the interface ID in Hello packets. Intra-Area-Prefix LSAs that reference Network LSAs have the Network LSAs LSID in the Referenced LSID field, so they will also be updated when the **interface-id snmp-if-index** command is entered. The old Network, Link, and Intra-Area-Prefix LSAs that are associated with a Network LSA will be flushed.

For both OSPFv2 and OSPFv3, adjacencies are not flapped, except for affected OSPFv3 demand circuits (including virtual links) with full adjacencies.

For both OSPFv2 and OSPFv3, if an interface does not have an SNMP ifIndex number and an interface ID is needed (for OSPFv2 this applies only to unnumbered interfaces and sham links), an error message will be generated and the interface will be disabled. The interface will be reenabled if the **no interface-id snmp-if-index** command is entered.

How to Configure SNMP ifIndex Value for Interface ID in Data Fields

Configuring OSPF interfaces to use SNMP MIB-II ifIndex Numbers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **router ospf** *process-id* [**vrf** *vpn-name*]
   -
   - **ipv6 router ospf** *process-id*
4. **interface-id snmp-if-index**

5. **end**
6. **show snmp mib ifmib ifindex** [*type number*] [**detail**][**free-list**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **router ospf** *process-id* [**vrf** *vpn-name*]
	•
	• **ipv6 router ospf** *process-id*
	Example:
	`Device(config)# router ospf 4`
	Example:
	Example:
	`Device(config)# ipv6 router ospf 4`
		Configures an OSPFv2 routing process and enters router configuration mode.
		Configures an OSPFv3 routing process and enters router configuration mode.
		Note If you configure an OSPFv3 routing process, that uses IPv6, you must have already enabled IPv6.

Step 4	**interface-id snmp-if-index** **Example:** `Device(config-router)# interface-id snmp-if-index`	Configures OSPF interfaces with the SNMP interface index identification numbers (ifIndex values).
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode. Repeat this task for each OSPF process for which you want the interfaces to use the SNMP MIB-II ifIndex numbers.
Step 6	**show snmp mib ifmib ifindex** [*type number*] [**detail**][**free-list**] **Example:** `Device# show snmp mib ifmib ifindex GigabitEtherent 0/0`	Displays SNMP interface index identification numbers (ifIndex values) for all the system interfaces or the specified system interface.

Configuration Examples for SNMP ifIndex Value for Interface ID in Data Fields

Example Configuring SNMP ifIndex Value for Interface ID for OSPFv2

The following example configures the OSPF interfaces to use the SNMP ifIndex values for the interfaces IDs. The **show snmp mib ifmib ifindex** command confirms that the SNMP MIB-II ifIndex values are used for the interface ID values in the OSPFv2 data fields.

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# router ospf 1
Device(config-router)# interface-id snmp-if-index
Device(config-router)# ^Z
Device# show ip ospf 1 1 data router self
OSPF Router with ID (172.16.0.1) (Process ID 1)
Router Link States (Area 1)
LS age: 6
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 172.16.0.1
Advertising Router: 172.16.0.1
LS Seq Number: 80000007
Checksum: 0x63AF
Length: 48
Area Border Router
Number of Links: 2
Link connected to: another Router (point-to-point)
(Link ID) Neighboring Router ID: 172.17.0.1
(Link Data) Router Interface address: 0.0.0.53
Number of TOS metrics: 0
TOS 0 Metrics: 64
Link connected to: a Stub Network
(Link ID) Network/subnet number: 192.168.0.11
(Link Data) Network Mask: 255.255.255.255
Number of TOS metrics: 0
TOS 0 Metrics: 1
Device# show snmp mib ifmib ifindex serial 13/0

Serial13/0: Ifindex = 53
```

Example Configuring SNMP ifIndex Value for Interface ID for OSPFv3

The following example configures the OSPFv3 interfaces to use the SNMP ifIndex values for the interface IDs:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ipv6 router ospf 1
Device(config-router)# interface-id snmp-if-index
```

The output from the **show snmp mib ifmib ifindex** command confirms that the SNMP MIB-II ifIndex values are being used for the interface ID values in the OSPFv2 data fields:

```
Device# show snmp mib ifmib ifindex GigabitEthernet 0/0/0
0/0/0: Ifindex = 5
Device# show ipv6 ospf interface
OSPF_VL0 is up, line protocol is up
  Interface ID 71
  Area 0, Process ID 1, Instance ID 0, Router ID 172.16.0.1
  Network Type VIRTUAL_LINK, Cost: 10
  Configured as demand circuit.
  Run as demand circuit.
  DoNotAge LSA allowed.
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Index 1/2/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.0.0.1  (Hello suppressed)
  Suppress hello for 1 neighbor(s)
GigabitEthernet is up, line protocol is up
  Link Local Address FE80::A8BB:CCFF:FE00:6F02, Interface ID 10
  Area 0, Process ID 1, Instance ID 0, Router ID 172.16.0.1
  Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.0.1, local address FE80::A8BB:CCFF:FE00:6F02
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:06
  Index 1/1/2, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
GigabitEthernet is up, line protocol is up
  Link Local Address FE80::A8BB:CCFF:FE00:6F01, Interface ID 6
  Area 1, Process ID 1, Instance ID 2, Router ID 172.16.0.1
  Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.0.1, local address FE80::A8BB:CCFF:FE00:6F01
  Backup Designated router (ID) 10.0.0.1, local address FE80::A8BB:CCFF:FE00:6E01
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:06
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.0.0.1  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
Device# show ipv6 ospf database network adv-router 172.16.0.1
OSPFv3 Router with ID (172.16.0.1) (Process ID 1)
Net Link States (Area 1)
  LS age: 144
  Options: (V6-Bit E-Bit R-bit DC-Bit)
  LS Type: Network Links
  Link State ID: 6 (Interface ID of Designated Router)
  Advertising Router: 172.16.0.1
  LS Seq Number: 80000001
  Checksum: 0x1FC0
  Length: 32
        Attached Router: 172.16.0.1
```

```
        Attached Router: 10.0.0.1
Device# show ipv6 ospf database prefix adv-router 172.16.0.1
OSPFv3 Router with ID (172.16.0.1) (Process ID 1)
Intra Area Prefix Link States (Area 0)
Routing Bit Set on this LSA
LS age: 196
LS Type: Intra-Area-Prefix-LSA
Link State ID: 0
Advertising Router: 172.16.0.1
LS Seq Number: 80000001
Checksum: 0x6F11
Length: 44
  Referenced LSA Type: 2001
  Referenced Link State ID: 0
  Referenced Advertising Router: 172.16.0.1
  Number of Prefixes: 1
  Prefix Address: 2002:0:2::
  Prefix Length: 64, Options: None, Metric: 10
Intra Area Prefix Link States (Area 1)
  Routing Bit Set on this LSA
  LS age: 161
  LS Type: Intra-Area-Prefix-LSA
  Link State ID: 0
  Advertising Router: 172.16.0.1
  LS Seq Number: 80000001
  Checksum: 0xB6E7
  Length: 52
  Referenced LSA Type: 2001
  Referenced Link State ID: 0
  Referenced Advertising Router: 172.16.0.1
  Number of Prefixes: 1
  Prefix Address: 2002:0:2:0:A8BB:CCFF:FE00:6F02
  Prefix Length: 128, Options: LA , Metric: 0
  Routing Bit Set on this LSA
  LS age: 151
  LS Type: Intra-Area-Prefix-LSA
  Link State ID: 1006
  Advertising Router: 172.16.0.1
  LS Seq Number: 80000001
  Checksum: 0x6E24
  Length: 44
  Referenced LSA Type: 2002
  Referenced Link State ID: 6
  Referenced Advertising Router: 172.16.0.1
  Number of Prefixes: 1
  Prefix Address: 2002:0:1::
  Prefix Length: 64, Options: None, Metric: 0
Device# show ipv6 ospf database router
OSPFv3 Router with ID (10.0.0.1) (Process ID 1)
Router Link States (Area 0)
  Routing Bit Set on this LSA
  LS age: 5 (DoNotAge)
  Options: (V6-Bit E-Bit R-bit DC-Bit)
  LS Type: Router Links
  Link State ID: 0
  Advertising Router: 10.0.0.1
  LS Seq Number: 80000004
  Checksum: 0xEE5C
  Length: 40
  Area Border Router
  Number of Links: 1
    Link connected to: a Virtual Link
      Link Metric: 10
      Local Interface ID: 70
```

```
      Neighbor Interface ID: 71
      Neighbor Router ID: 172.16.0.1
LS age: 162
Options: (V6-Bit E-Bit R-bit DC-Bit)
LS Type: Router Links
Link State ID: 0
Advertising Router: 172.16.0.1
LS Seq Number: 80000004
Checksum: 0xCE7C
Length: 40
Area Border Router
Number of Links: 1
    Link connected to: a Virtual Link
      Link Metric: 10
      Local Interface ID: 71
      Neighbor Interface ID: 70
      Neighbor Router ID: 10.0.0.1
Router Link States (Area 1)
Routing Bit Set on this LSA
LS age: 176
Options: (V6-Bit E-Bit R-bit DC-Bit)
LS Type: Router Links
Link State ID: 0
Advertising Router: 10.0.0.1
LS Seq Number: 80000003
Checksum: 0xC807
Length: 40
Area Border Router
Number of Links: 1
    Link connected to: a Transit Network
Link Metric: 10
Local Interface ID: 6
Neighbor (DR) Interface ID: 6
Neighbor (DR) Router ID: 172.16.0.1
LS age: 175
Options: (V6-Bit E-Bit R-bit DC-Bit)
LS Type: Router Links
Link State ID: 0
Advertising Router: 172.16.0.1
LS Seq Number: 80000004
Checksum: 0xBD10
Length: 40
Area Border Router
Number of Links: 1
    Link connected to: a Transit Network
Link Metric: 10
Local Interface ID: 6
Neighbor (DR) Interface ID: 6
Neighbor (DR) Router ID: 172.16.0.1
Device# show ipv6 ospf database link adv-router 172.16.0.1
OSPFv3 Router with ID (172.16.0.1) (Process ID 1)
Link (Type-8) Link States (Area 0)
  LS age: 245
  Options: (V6-Bit E-Bit R-bit DC-Bit)
  LS Type: Link-LSA (Interface: GigabitEthernet2/0)
  Link State ID: 10 (Interface ID)
  Advertising Router: 172.16.0.1
  LS Seq Number: 80000002
  Checksum: 0xA0CB
  Length: 56
  Router Priority: 1
  Link Local Address: FE80::A8BB:CCFF:FE00:6F02
  Number of Prefixes: 1
  Prefix Address: 2002:0:2::
```

```
  Prefix Length: 64, Options: None
Link (Type-8) Link States (Area 1)
LS age: 250
Options: (V6-Bit E-Bit R-bit DC-Bit)
LS Type: Link-LSA (Interface: GigabitEthernet1/0)
Link State ID: 6 (Interface ID)
Advertising Router: 172.16.0.1
LS Seq Number: 80000001
Checksum: 0x4F94
Length: 44
Router Priority: 1
Link Local Address: FE80::A8BB:CCFF:FE00:6F01
Number of Prefixes: 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
Protecting TE tunnel interfaces	MPLS Traffic Engineering--Fast Reroute Link and Node Protection section in the *Cisco IOS Multiprotocol Label Switching Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
•	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 5286	Basic Specification for IP Fast Reroute: Loop-Free Alternates

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF SNMP ifIndex Value for Interface ID

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 274: Feature Information for OSPF: SNMP ifIndex Value for Interface ID in OSPFv2 and OSPFv3 Data Fields

Feature Name	Releases	Feature Information
OSPF: SNMP ifIndex Value for Interface ID in OSPFv2 and OSPFv3 Data Fields	Cisco IOS XE Release 2.6	This allows you to choose either the current interface number or the SNMP ifIndex value for the interface ID in OSPFv2 and OSPFv3 data fields. The advantage to using the SNMP MIB-II ifIndex value is that this number corresponds to the number that the user will see reported by SNMP. The following command is introduced or modified by the feature documented in this module: **interface-id snmp-if-index**

Table 275: Feature Information for OSPF: SNMP ifIndex Value for Interface ID in OSPFv2 and OSPFv3 Data Fields

Feature Name	Releases	Feature Information
OSPF: SNMP ifIndex Value for Interface ID in OSPFv2 and OSPFv3 Data Fields	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 232

OSPFv2 Local RIB

With the OSPFv2 Local RIB feature, each OSPF protocol instance has its own local Routing Information Base (RIB). The OSPF local RIB serves as the primary state for OSPF SPF route computation. The global RIB is not updated with intermediate results during the SPF. Instead, the global RIB is updated only when routes are added, deleted, or changed, thereby reducing global RIB computation. This reduced update activity may result in fewer dropped packets.

This feature is enabled by default and does not need to be configured. This document describes some optional configuration tasks to modify how the global and local RIBs function, although it is recommended to keep the default settings.

Prerequisites for OSPFv2 Local RIB

Before this feature is configured, the OSPF routing protocol must be configured.

Restrictions for OSPFv2 Local RIB

This feature is available only for IP Version 4 networks.

Information About OSPFv2 Local RIB

A router that is running OSPFv2 maintains a local RIB in which it stores all routes to destinations that it has learned from its neighbors. At the end of each SPF, OSPF attempts to install the best (that is, the least-cost) routes to a destination present in the local RIB into the global IPv4 routing table. The global RIB will be updated only when routes are added, deleted, or changed. Routes in the local RIB and Forwarding Information Base (FIB) will not compute when intermediate results are computed during SPF, resulting in fewer dropped packets in some circumstances.

By default, the contents of the global RIB are used to compute inter-area summaries, NSSA translation, and forwarding addresses for type-5 and type-7 LSAs. Each of these functions can be configured to use the contents of the OSPF local RIB instead of the global RIB for their computation. Using the local RIB for the computation may be slightly faster in some circumstances, but because the local RIB has information for only a particular instance of OSPF, using it for the computation may yield incorrect results. Potential problems that may occur include routing loops and null routes. It is recommended that you not change the default values because they are conservative and preserve the current global RIB behavior.

By default, OSPF installs discard routes to null0 for any area range (internal) or summary-address (external) prefixes that it advertises to other routers. Installation of a discard route can prevent routing loops in cases where portions of a summary do not have a more specific route in the RIB. Normally, internal discard routes are installed with an administrative distance of 110, while external discard routes have an administrative distance of 254.

There may be rare circumstances, however, when some other values are needed. For example, if one OSPF process installs a route that exactly matches an area range configured on another OSPF process, the internal discard routes for the second OSPF process could be given a higher (less desirable) administrative distance.

How to Configure OSPFv2 Local RIB

Although it is recommended to keep the default settings for the commands described in the following sections, it is optional to change the defaults settings.

Changing the Default Local RIB Criteria

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **local-rib-criteria** [**forwarding-address**] [**inter-area-summary**] [**nssa-translation**]
5. **end**
6. **show ip ospf** *process-id* **rib** [**redistribution**] [*network-prefix*] [*network-mask*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# router ospf 23`	Configures an OSPFv2 routing process and enters router configuration mode.
Step 4	**local-rib-criteria** [**forwarding-address**] [**inter-area-summary**] [**nssa-translation**] **Example:** `Device(config-router)# local-rib-criteria forwarding-address`	Specifies that the OSPF local RIB will be used for route validation.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 6	**show ip ospf** *process-id* **rib** [**redistribution**] [*network-prefix*] [*network-mask*] [**detail**] **Example:** `Device# show ip ospf 23 rib`	Displays information for the OSPF local RIB or locally redistributed routes.

Changing the Administrative Distance for Discard Routes

Note It is recommended that you keep the default settings. However, you can follow the steps in this section to change the administrative distance for discard routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vpn-name*]
4. **discard-route** [**external** [*distance*]] [**internal** [*distance*]]
5. **end**
6. **show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# router ospf 23`	Configures an OSPFv2 routing process and enters router configuration mode.
Step 4	**discard-route** [**external** [*distance*]] [**internal** [*distance*]] **Example:** `Device(config-router)# discard-route external 150`	Reinstalls either an external or internal discard route that was previously removed. **Note** You can now specify the administrative distance for internal and external discard routes.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 6	**show ip route** [*ip-address* [*mask*] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**] **Example:** `Device# show ip route ospf 23`	Displays the current state of the routing table. **Note** Entering the **show ip route** command will verify the changed administrative distance values for external and internal discard routes.

Example

The sample output displayed for the **show ip route** command confirms that the administrative distance for the IP route 192.168.0.0/24 is 110.

```
Device# show ip route 192.168.0.0 255.255.255.0

Routing entry for 192.168.0.0/24

        Known via "ospf 1", distance 110, metric 0, type intra area

  Routing Descriptor Blocks:

  * directly connected, via Null0
```

```
        Route metric is 0, traffic share count is 1
```

Troubleshooting Tips

You can research the output from the **debug ip ospf rib** command to learn about the function of the local RIB and the interaction between the route redistribution process and the global RIB. For example, you can learn why the routes that OSPF placed in the global RIB are not the same ones that you anticipated.

Configuration Examples for OSPFv2 Local RIB

Example: Changing the Default Local RIB Criteria

In the following example, the **local-rib-criteria** command is entered without any keywords to specify that the local RIB will be used as criteria for all of the following options: forwarding address, inter-area summary, and NSSA translation.

```
router ospf 1
 router-id 10.0.0.6
 local-rib-criteria
```

Example: Changing the Administrative Distance for Discard Routes

In the following example, the administrative distance for external and internal discard routes is set to 25 and 30, respectively.

```
router ospf 1
 router-id 10.0.0.6
 log-adjacency-changes
 discard-route external 25 internal 30
 area 4 range 10.2.0.0 255.255.0.0
 summary-address 192.168.130.2 255.255.255.0
 redistribute static subnets
 network 192.168.129.2 0.255.255.255 area 0
 network 192.168.130.12 0.255.255.255 area 0
```

The output from the **show ip route** command verifies that the administrative distance for the internal route 10.2.0.0/16 is set to 30.

```
Device# show ip route 10.2.0.0 255.255.0.0
Routing entry for 10.2.0.0/16
Known via "ospf 1", distance 30, metric 1, type intra area
  Routing Descriptor Blocks:
  * directly connected, via Null0
      Route metric is 1, traffic share count is 1
```

The output from the **show ip route** command verifies that the administrative distance for the external route 192.168.130.2/24 is set to 25.

```
Device# show ip route 192.168.130.2 255.255.255.0
Routing entry for 192.168.130.2/24
 Known via "ospf 1", distance 25, metric 20, type intra area
```

```
Routing Descriptor Blocks:
* directly connected, via Null0
    Route metric is 20, traffic share count is 1
```

Additional References

The following sections provide references related to OSPFv2 Local RIB.

Related Documents

Related Topic	Document Title
Configuring OSPF	Configuring OSPF
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv2 Local RIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 276: Feature Information for the OSPFv2 Local RIB

Feature Name	Releases	Feature Information
OSPFv2 Local RIB	Cisco IOS XE Release 2.1	With the OSPFv2 Local RIB feature, each OSPF protocol instance has its own local Routing Information Base (RIB). The OSPF local RIB serves as the primary state for OSPF SPF route computation. The global RIB is not updated with intermediate results during the SPF. Instead, the global RIB is updated only when routes are added, deleted, or changed, thereby reducing global RIB computation. This reduced update activity may result in fewer dropped packets. This feature is enabled by default and does not need to be configured. This document describes some optional configuration tasks to modify how the global and local RIBs function, although it is recommended to keep the default settings. The following commands were introduced or modified: **debug ip ospf rib**, **discard-route**, **local-rib-criteria**, **show ip ospf rib**.

Table 277: Feature Information for the OSPFv2 Local RIB

Feature Name	Releases	Feature Information
OSPFv2 Local RIB	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 233

OSPF Support for Forwarding Adjacencies over MPLS TE Tunnels

The OSPF Support for Forwarding Adjacencies over MPLS Traffic Engineered Tunnels feature adds Open Shortest Path First (OSPF) support to the Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Forwarding Adjacency feature, which allows a network administrator to handle a traffic engineering, label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network based on the shortest path first (SPF) algorithm. An OSPF forwarding adjacency can be created between routers in the same area.

History for the OSPF Support for Forwarding Adjacencies over MPLS Traffic Engineered Tunnels Feature

Prerequisites for OSPF Forwarding Adjacency

- OSPF must be configured in your network.
- Cisco Express Forwarding (CEF) must be enabled.
- You should understand MPLS TE tunnels for forwarding adjacency as described in the " MPLS Traffic Engineering Forwarding Adjacency" module.

Information About OSPF Forwarding Adjacency

OSPF includes MPLS TE tunnels in the OSPF link-state database in the same way that other links appear for purposes of routing and forwarding traffic. When an MPLS TE tunnel is configured between networking devices, that link is considered a forwarding adjacency. The user can assign a cost to the tunnel to indicate the link's preference. Other networking devices will see the tunnel as a link in addition to the physical link.

How to Configure OSPF Forwarding Adjacency

Configuring OSPF Forwarding Adjacency

Note Configure a forwarding adjacency on two LSP tunnels bidirectionally, from A to B and B to A. Otherwise, the forwarding adjacency is advertised, but not used in the IGP network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef distributed**
4. **mpls traffic-eng tunnels**
5. **interface loopback** *number*
6. **ip address** *ip-address mask*
7. **no shutdown**
8. **exit**
9. **interface tunnel** *number*
10. **tunnel mode mpls traffic-eng**
11. **tunnel mpls traffic-eng forwarding-adjacency** {**holdtime** *value*}
12. **ip ospf cost** *cost*
13. **exit**
14. **router ospf** *process-id*
15. **mpls traffic-eng router-id** *interface*
16. **mpls traffic-eng area** *number*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip cef distributed** **Example:**	Enables Cisco Express Forwarding (CEF).

	Command or Action	Purpose
	`Router(config)# ip cef distributed`	
Step 4	**mpls traffic-eng tunnels** **Example:** `Router(config)# mpls traffic-eng tunnels`	Enables MPLS traffic engineering tunnel signaling on a device.
Step 5	**interface loopback** *number* **Example:** `Router(config)# interface loopback0`	Configures a loopback interface and enters interface configuration mode. • Set up a loopback interface with a 32-bit mask, enable CEF, enable MPLS traffic engineering, and set up a routing protocol (OSPF) for the MPLS network.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.255`	Configures the IP address and subnet mask of the loopback interface.
Step 7	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 1`	Designates a tunnel interface for the forwarding adjacency and enters interface configuration mode.
Step 10	**tunnel mode mpls traffic-eng** **Example:** `Router(config-if)# tunnel mode mpls traffic-eng`	Sets the mode of a tunnel to MPLS for traffic engineering.
Step 11	**tunnel mpls traffic-eng forwarding-adjacency** {**holdtime** *value*} **Example:** `Router(config-if)# tunnel mpls traffic-eng forwarding-adjacency holdtime 10000`	Advertises a TE tunnel as a link in an IGP network. • The **holdtime** *value* keyword argument combination is the time in milliseconds (ms) that a TE tunnel waits after going down before informing the network. The range is 0 to 4,294,967,295 ms. The default value is 0.

	Command or Action	Purpose
Step 12	**ip ospf cost** *cost* **Example:** `Router(config-if)# ip ospf cost 4`	(Optional) Configures the cost metric for a tunnel interface to be used as a forwarding adjacency.
Step 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 14	**router ospf** *process-id* **Example:** `Router(config)# router ospf 1`	Configures an OSPF routing process and enters router configuration mode.
Step 15	**mpls traffic-eng router-id** *interface* **Example:** `Router(config-router)# mpls traffic-eng router-id ethernet 1/0`	Specifies that the traffic engineering router identifier for the node is the IP address associated with a given interface.
Step 16	**mpls traffic-eng area** *number* **Example:** `Router(config-router)# mpls traffic-eng area 1`	Configures a router running OSPF MPLS so that it floods traffic engineering for the indicated OSPF area.
Step 17	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode.

Configuration Examples for OSPF Forwarding Adjacency

Example OSPF Forwarding Adjacency

In the following example, the tunnel destination is the loopback interface on the other router. The router is configured with OSPF TE extensions and it floods traffic engineering link-state advertisements (LSAs) in OSPF area 0. The traffic engineering router identifier for the node is the IP address associated with Loopback 0. The last five lines of the example set up the routing protocol for the MPLS network, which is OSPF in this case.

Note Do not use the **mpls traffic-eng autoroute announce** command if you configure a forwarding adjacency in the tunnel.

```
ip routing
ip cef distributed
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 127.0.0.1 255.255.255.255
 no shutdown
!
interface Tunnel1
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 10.1.1.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng forwarding-adjacency holdtime 10000
 ip ospf cost 4
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth  10
 tunnel mpls traffic-eng path-option 2 dynamic
router ospf 5
 log-adjacency-changes
 network 10.1.1.1 0.0.0.0 area 0
 mpls traffic-eng router-id loopback0
 mpls traffic-eng area 0
```

When you look at the self-generated router LSA, you will see it as one of the links in router LSA (shown in bold in the following output).

```
Router# show ip ospf database route self-originate
OSPF Router with ID (10.5.5.5) (Process ID 5)
                Router Link States (Area 0)
  LS age:332
  Options:(No TOS-capability, DC)
  LS Type:Router Links
  Link State ID:10.5.5.5
  Advertising Router:10.5.5.5
  LS Seq Number:80000004
  Checksum:0x1D24
  Length:72
  Number of Links:4
    Link connected to another Router (point-to-point)
     (Link ID) Neighboring Router ID:10.3.3.3
     (Link Data) Router Interface address:0.0.0.23
      Number of TOS metrics:0
       TOS 0 Metrics:1562
    Link connected to:a Transit Network
     (Link ID) Designated Router address:172.16.0.1
     (Link Data) Router Interface address:172.16.0.2
      Number of TOS metrics:0
       TOS 0 Metrics:10
    Link connected to:a Transit Network
     (Link ID) Designated Router address:172.16.0.3
     (Link Data) Router Interface address:172.16.0.4
      Number of TOS metrics:0
       TOS 0 Metrics:10
    Link connected to:a Stub Network
     (Link ID) Network/subnet number:10.5.5.5
     (Link Data) Network Mask:255.255.255.255
      Number of TOS metrics:0
       TOS 0 Metrics:1
```

Additional References

The following sections provide references related to OSPF Forwarding Adjacency.

Related Documents

Related Topic	Document Title
MPLS traffic engineering forwarding adjacency	MPLS Traffic Engineering Forwarding Adjacency
Configuring OSPF for MPLS traffic engineering	MPLS Traffic Engineering and Enhancements
MPLS Traffic Engineering - LSP Attributes	MPLS Traffic Engineering - LSP Attributes

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 234

Enabling OSPFv2 on an Interface Basis

This document describes how to enable Open Shortest Path First version 2 (OSPFv2) on a per-interface basis to simplify the configuration of unnumbered interfaces. The **ip ospf area** command allows you to enable OSPFv2 explicitly on an interface. The **ip ospf area** command is an alternative to enabling OSPFv2 through the address of the interface that matches the address range specified by the **network area** command.

Prerequisites for Enabling OSPFv2 on an Interface Basis

OSPFv2 must be running on your network.

Restrictions on Enabling OSPFv2 on an Interface Basis

The **ip ospf area** command is supported only for OSPFv2.

Information About Enabling OSPFv2 on an Interface Basis

Benefits of Enabling OSPFv2 on an Interface Basis

OSPF is enabled on an interface when the network address for the interface matches the range of addresses that is specified by the **network area** command, which is entered in router configuration mode. Alternatively, you can enable OSPFv2 explicitly on an interface by using the **ip ospf area** command, which is entered in interface configuration mode. This capability simplifies the configuration of unnumbered interfaces with different areas.

Because the **ip ospf area** command is configured explicitly for an interface, it supersedes the effects of the **network area**command, which is entered at the network level to affect the interfaces whose addresses fall within the address range specified for the **network area** command.

If you later disable the **ip ospf area** command, the interface still will run OSPFv2 as long as its network address matches the range of addresses that is specified by the **network area**command.

Implications of Configuring OSPFv2 On a Router Basis or an Interface Basis

Before you use the **ip ospf area** command to enable OSPFv2 on an interface, we recommend that you understand the following scenarios and command behavior. There are implications to using the **network area**command (configuring OSPFv2 in router configuration mode) versus using the **ip ospf area** command (configuring OSPFv2 in interface configuration mode).

Interface Is Already OSPFv2-Enabled by network area Command with Same Area and Process

If you enter the **ip ospf area** command on an interface that is enabled in OSPFv2 by the **network area**command, the process ID or area ID of the interface does not change, and the interface status will not be changed. However, the interface will be flagged as being configured from interface configuration mode, and the configuration data will be saved in the interface description block (IDB).

Interface Is Already Configured by network area Command with Different Area or Process

If you enter the **ip ospf area** command on an interface that is enabled in OSPFv2 by the **network area**command, but you change the configuration by changing the process ID and area ID of the interface, after the new configuration information is stored in the IDB, the interface will be removed and reattached. Therefore, the interface will be removed from the original area and process and be added to the new ones. The state of the interface will also be reset.

Interface Is Not Configured by network area Command

If the interface is not enabled in OSPFv2 by the **network area** command, the area and OSPF router instance will be created if needed. When the router is reloaded, the OSPF process will not begin running until system initialization is complete. To remove an OSPF router instance, enter the **no router ospf** command. Removing the **ip ospf area** command in interface mode will not result in removing an OSPF router instance.

Removing an ip ospf area Command

When the **ip ospf area**command is removed, the interface will be detached from the area. The area will be removed if it has no other attached interfaces. If the interface address is covered by the **network area** command, the interface will be enabled once again in the area for the network that it is in.

New Processes

If an OSPF process does not already exist, and a router ID cannot be chosen when either the **router ospf** command or the **interface** command is configured, a Proximity Database (PDB) and a process will be created, but the process will be inactive. The process will become active when a router ID is chosen, either when it is explicitly configured using the **router-id** command or when an IP address becomes available. Note that the **router ospf** command will now be accepted even if a router ID cannot be chosen, putting the command-line interface (CLI) into the OSPF configuration context. Therefore, the **router-id** command is to be entered before an IP address is available. If the process is not active and the **show ip ospf**command is entered, the message "%OSPF: Router process X is not running, please provide a router-id" will be displayed.

Link-State Advertisements and Shortest Path First

If a state change occurs as a result of the **ip ospf area**command, new router link-state advertisements (LSAs) will be generated (also for the old area, if the interface is changing areas) and shortest path first (SPF) will be scheduled to run in both the old and new areas.

How to Enable OSPFv2 on an Interface Basis

Enabling OSPFv2 on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf** *process-id* **area** *area-id* [**secondaries none**]
5. **end**
6. **show ip ospf interface** [*type -number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/2/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip ospf** *process-id* **area** *area-id* [**secondaries none**] **Example:** `Device(config-if)# ip ospf 1 area 0 secondaries none`	Enables OSPFv2 on an interface. • To prevent secondary IP addresses on the interface from being advertised, you must enter the optional **secondaries** keyword followed by the **none** keyword.
Step 5	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
Step 6	**show ip ospf interface** [*type -number*] **Example:** `Device# show ip ospf interface GigabitEthernet 0/2/1`	Displays OSPF-related interface information. • Once you have enabled OSPFv2 on the interface, you can enter the **show ip ospf interface**command to verify the configuration.

Configuration Example for Enabling OSPFv2 on an Interface

Example Enabling OSPFv2 on an Interface

In the following example, OSPFv2 is configured explicitly on GigabitEthernet interface 0/0/0:

```
Device(config)# interface GigabitEthernet 0/2/1
Device(config-if)# bandwidth 10000
Device(config-if)# ip address 172.16.1.1 255.255.255.0
Device(config-if)# ip ospf hello-interval 1
Device(config-if)# ip ospf 1 area 0
```

When the **show ip ospf interface** command is entered, the following output shows that GigabitEthernet interface 0/0/0 was configured in interface configuration mode to run OSPFv2. The secondary IP addresses on the interface will also be advertised:

```
Device# show ip ospf interface GigabitEthernet 0/2/1
GigabitEthernet0/0/0 is up, line protocol is up
  Internet Address 172.16.1.1/24, Area 0
  Process ID 1, Router ID 172.16.11.11, Network Type BROADCAST, Cost: 10
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.11.11, Interface address 172.16.1.1
  Backup Designated router (ID) 172.16.22.11, Interface address 172.16.1.2
  Timer intervals configured, Hello 1, Dead 4, Wait 4, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
  Supports Link-local Signaling (LLS)
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 172.26.22.11  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

Additional References

The following sections provide references related to enabling OSPFv2 on an interface.

Related Documents

Related Topic	Document Title
Configuring OSPF	Configuring OSPF
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2328	*OSPF Version 2*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Enabling OSPFv2 on an Interface Basis

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 278: Feature Information for Enabling OSPFv2 on an Interface Basis

Feature Name	Releases	Feature Information
Enabling OSPFv2 on an Interface Basis **Note** This feature was originally named "Area Command in Interface Mode for OSPFv2."	Cisco IOS XE Release 2.1	This document describes how to enable OSPFv2 on a per-interface basis to simplify the configuration of unnumbered interfaces. The **ip ospf area** command allows you to enable OSPFv2 explicitly on an interface. The **ip ospf area** command is an alternative to enabling OSPFv2 through the address of the interface that matches the address range specified by the **network area** command. The following commands are introduced or modified in the feature documented in this module: • **ip ospf area.**

Table 279: Feature Information for Enabling OSPFv2 on an Interface Basis

Feature Name	Releases	Feature Information
Enabling OSPFv2 on an Interface Basis **Note** This feature was originally named "Area Command in Interface Mode for OSPFv2."	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 235

OSPF Nonstop Routing

The OSPF Nonstop Routing feature allows a device with redundant Route Processors (RPs) to maintain its Open Shortest Path First (OSPF) state and adjacencies across planned and unplanned RP switchovers. The OSPF state is maintained by checkpointing the state information from OSPF on the active RP to the standby RP. After a switchover to the standby RP, OSPF uses the checkpointed information to continue operations without interruption.

Prerequisites for OSPF NSR

- OSPF NSR is available for platforms with redundant RPs or Cisco IOS software redundancy running Cisco IOS Release XE 3.3S or later releases.

Restrictions for OSPF NSR

- OSPF nonstop routing (NSR) can significantly increase the memory used by OSPF during certain phases of its operation. CPU usage also can be increased. You should be aware of router memory capacity and estimate the likely memory requirements of OSPF NSR. For more information see Configuring OSPF NSR. For routers where memory and CPU are constrained you might want to consider using OSPF NSF instead. For more information, see OSPF RFC 3623 Graceful Restart Helper Mode.
- A switchover from the active to the standby RP can take several seconds, depending on the hardware platform, and during this time OSPF is unable to send Hello packets. As a result, configurations that use small OSPF dead intervals might not be able to maintain adjacencies across a switchover.

Information About OSPFv3 Authentication Trailer

OSPF NSR Functionality

Although OSPF Nonstop Routing (NSR) serves a similar function to OSPF Nonstop Forwarding (NSF), it works differently. With NSF, OSPF on the newly active standby RP initially has no state information. OSPF uses extensions to the OSPF protocol to recover its state from neighboring OSPF devices. For the recovery to work, the neighbors must support the NSF protocol extensions and be willing to act as “helpers” to the device that is restarting. The neighbors must also continue forwarding data traffic to the device that is restarting while protocol state recovery takes place.

With NSR, by contrast, the device that performs the switchover preserves its state internally, and in most cases the neighbors are unaware of the switchover. Because assistance is not needed from neighboring devices, NSR can be used in situations where NSF cannot be used; for example, in networks where not all neighbors implement the NSF protocol extensions, or where network topology changes during the recovery making NSF unreliable, use NSR instead of NSF.

How to Configure OSPF Nonstop Routing

Configuring OSPF NSR

Perform this task to configure OSPF NSR.

NSR adds a single new line, "nsr," to the OSPF router mode configuration. Routers that do not support NSR, for whatever reason, will not accept this command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **nsr**
5. **end**
6. **show ip ospf [** *process-id* **] nsr [[objects]|[statistics]]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 109`	Places the router in router configuration mode and configures an OSPF routing process.
Step 4	**nsr** **Example:** `Router(config-router)# nsr`	Configures NSR.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 6	**show ip ospf [** *process-id* **] nsr [[objects]|[statistics]]** **Example:** `Router# show ip ospf 109 nsr`	Displays OSPF NSR status information.

Troubleshooting Tips

OSPF NSR can increase the amount of memory used by the OSPF device process. To determine how much memory OSPF is currently using without NSR, you can use the **show processes** and **show processes memory** commands:

```
Device# show processes|include OSPF

 276 Mwe  133BE14         1900       1792    1060 8904/12000  0 OSPF-1 Router
 296 Mwe  133A824           10        971      10 8640/12000  0 OSPF-1 Hello
```

Process 276 is the OSPF device process that is to be checked. Use the **show processes memory** command to display its current memory use:

```
Device# show processes memory 276

Process ID: 276
Process Name: OSPF-1 Router
Total Memory Held: 4454800 bytes
```

In the above example, OSPF is using 4,454,800 bytes, or approximately 4.5 megabytes (MB). Because OSPF NSR can consume double this memory for brief periods, ensure that the device has at least 5 MB of free memory before enabling OSPF NSR.

Configuration Examples for OSPF Nonstop Routing

Example: Configuring OSPF NSR

The following example shows how to configure OSPF NSR:

```
Device> enable
Device# configure terminal
Device(config)# router ospf 1
Device(config-router)# nsr
Device(config-router)# end
Device# show ip ospf 1 nsr
Standby RP
 Operating in duplex mode
 Redundancy state: STANDBY HOT
 Peer redundancy state: ACTIVE
 ISSU negotation complete
 ISSU versions compatible
Routing Process "ospf 1" with ID 10.1.1.100
 NSR configured
 Checkpoint message sequence number: 3290
 Standby synchronization state: synchronized
 Bulk sync operations:  1
 Last sync start time:  15:22:48.971 UTC Fri Jan 14 2011
 Last sync finish time: 15:22:48.971 UTC Fri Jan 14 2011
 Last sync lost time:   -
 Last sync reset time:  -
 LSA Count: 2, Checksum Sum 0x00008AB4
```

The output shows that OSPF NSR is configured and that OSPF on the standby RP is fully synchronized and ready to continue operation should the active RP fail or if a manual switchover is performed.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF	"Configuring OSPF" in the *IP Routing: OSPF Configuration Guide*.
OSPFv2 loop-free alternate fast reroute	"OSPFv2 Loop-Free Alternate Fast Reroute" in the *IP Routing: OSPF Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 5286	*Basic Specification for IP Fast Reroute: Loop-Free Alternates*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF NSR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 280: Feature Information for OSPF NSR

Feature Name	Releases	Feature Information
OSPF NSR	XE 3.3S Cisco IOS Release 15.1(1)SY	The OSPF NSR feature allows a router with redundant route processors to maintain its OSPF state and adjacencies across planned and unplanned RP switchovers. In Cisco IOS Release XE 3.3S, this feature was introduced. The following commands were introduced or modified: **nsr, show ip ospf nsr**.

Table 281: Feature Information for OSPF NSR

Feature Name	Releases	Feature Information
OSPF NSR	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 236

OSPFv3 NSR

The OSPFv3 NSR feature allows a router with redundant Route Processors (RPs) to maintain its Open Shortest Path First (OSPF) state and adjacencies across planned and unplanned RP switchovers. It does this by checkpointing state information from OSPFv3 on the active RP to the standby RP. Later, following a switchover to the standby RP, OSPFv3 can use this checkpointed information to continue operation without interruption.

Information About OSPFv3 NSR

OSPFv3 NSR Functionality

Although OSPFv3 NSR serves a similar function to the OSPFv3 graceful restart feature, it works differently. With graceful restart, OSPFv3 on the newly active standby RP initially has no state information, so it uses extensions to the OSPFv3 protocol to recover its state from neighboring OSPFv3 devices. For this to work, the neighbors must support the graceful restart protocol extensions and be able to act as helpers to the restarting device. They must also continue forwarding data traffic to the restarting device while this recovery is taking place.

With NSR, by contrast, the device performing the switchover preserves its state internally, and in most cases the neighbors are unaware that anything has happened. Because no assistance is needed from neighboring devices, NSR can be used in situations where graceful restart cannot; for example, graceful restart is unreliable in networks where not all the neighbors implement the graceful restart protocol extensions or where the network topology changes during the recovery.

Note When NSR is enabled, the responsiveness and scalability of OSPF is degraded. The performance degradation happens because OSPF uses cpu and memory to checkpoint data to the standby Route Processor (RP).

How to Configure OSPFv3 NSR

Configuring OSPFv3 NSR

Perform this task to configure OSPFv3 NSR.

Note Devices that do not support NSR will not accept the **nsr** (OSPFv3) command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** *process-id*
4. **nsr**
5. **end**
6. **show ospfv3** [*process-id*] [*address-family*] **nsr**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** *process-id* **Example:** `Device(config)# router ospfv3 109`	Places the device in router configuration mode and configures an OSPFv3 routing process.
Step 4	**nsr** **Example:** `Device(config-router)# nsr`	Configures NSR.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ospfv3** [*process-id*] [*address-family*] **nsr** **Example:** `Device# show ospfv3 109 nsr`	Displays OSPFv3 NSR status information.

Configuring OSPFv3 NSR for an Address Family

In address family configuration mode you can configure NSR for a particular address family. Perform this task to enable OSPFv3 NSR for an address family.

Note Devices that do not support NSR will not accept the **nsr** (OSPFv3) command.

SUMMARY STEPS

1. **router ospfv3** *process-id*
2. **address-family** {**ipv4** | **ipv6**} **unicast** [**vrf** *vrf-name*]
3. **nsr** [**disable**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router ospfv3** *process-id* **Example:** `Device(config)# router ospfv3 109`	Places the device in router configuration mode and configures an OSPFv3 routing process.
Step 2	**address-family** {**ipv4**	**ipv6**} **unicast** [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv4 unicast`	Enters IPv4 or IPv6 address family configuration mode for OSPFv3 router configuration mode.
Step 3	**nsr** [**disable**] **Example:** `Device(config-router-af)# nsr`	Enables NSR for the address family that is configured.

Disabling OSPFv3 NSR for an Address Family

In address family configuration mode the optional **disable** keyword is available for the **nsr** command. Perform this task to disable OSPFv3 NSR for an address family.

SUMMARY STEPS

1. **router ospfv3** *process-id*

2. **address-family** {**ipv4** | **ipv6**} **unicast** [**vrf** *vrf-name*]
3. **nsr** [**disable**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router ospfv3** *process-id* **Example:** `Device(config)# router ospfv3 109`	Places the device in router configuration mode and configures an OSPFv3 routing process.
Step 2	**address-family** {**ipv4** | **ipv6**} **unicast** [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv6 unicast`	Enters IPv4 or IPv6 address family configuration mode for OSPFv3 router configuration mode.
Step 3	**nsr** [**disable**] **Example:** `Device(config-router-af)# nsr disable`	Disables NSR for the address family that is configured.

Troubleshooting Tips

OSPFv3 NSR can increase the amount of memory used by the OSPFv3 device process. To determine how much memory OSPFv3 is currently using without NSR, you can use the **show processes** and **show processes memory** commands:

```
Device# show processes
 | include OSPFv3
 276 Mwe  133BE14         1900       1792    1060 8904/12000  0 OSPFv3-1 Router
 296 Mwe  133A824           10        971      10 8640/12000  0 OSPFv3-1 Hello
```

Process 276 is the OSPFv3 device process that is to be checked. The **show processes memory** command is used to display its current memory use:

```
Device# show processes memory 276
Process ID: 276
Process Name: OSPFv3-1 Router
Total Memory Held: 4454800 bytes
```

In this case OSPFv3 is using 4,454,800 bytes or approximately 4.5 megabytes (MB). OSPFv3 NSR could double this for brief periods, so you should make sure the device has at least 5 MB of free memory before enabling OSPFv3 NSR.

Configuration Examples for OSPFv3 NSR

Example Configuring OSPFv3 NSR

The following example shows how to configure OSPFv3 NSR and verify that it is enabled:

```
Device(config)# router ospfv3 1
Device(config-router)# nsr
Device(config-router)# end
Device# show ospfv3 1
  OSPFv3 1 address-family ipv4
  Router ID 10.0.0.1
  Supports NSSA (compatible with RFC 3101)
  Event-log enabled, Maximum number of events: 1000, Mode: cyclic
  It is an area border and autonomous system boundary router
  Redistributing External Routes from,
  Router is not originating router-LSAs with maximum metric
  Initial SPF schedule delay 5000 msecs
  Minimum hold time between two consecutive SPFs 10000 msecs
  Maximum wait time between two consecutive SPFs 10000 msecs
  Minimum LSA interval 5 secs
  Minimum LSA arrival 1000 msecs
  LSA group pacing timer 240 secs
  Interface flood pacing timer 33 msecs
  Retransmission pacing timer 66 msecs
  Retransmission limit dc 24 non-dc 24
  Number of external LSA 0. Checksum Sum 0x000000
  Number of areas in this router is 3. 2 normal 0 stub 1 nssa
  Non-Stop Routing enabled
  Graceful restart helper support enabled
  Reference bandwidth unit is 100 mbps
  RFC1583 compatibility enabled
     Area BACKBONE(0) (Inactive)
         Number of interfaces in this area is 1
         SPF algorithm executed 3 times
         Number of LSA 6. Checksum Sum 0x03C938
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
     Area 1
         Number of interfaces in this area is 3
         SPF algorithm executed 3 times
         Number of LSA 6. Checksum Sum 0x024041
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
     Area 3
         Number of interfaces in this area is 1
         It is a NSSA area
         Perform type-7/type-5 LSA translation
         SPF algorithm executed 4 times
         Number of LSA 5. Checksum Sum 0x024910
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0

  OSPFv3 1 address-family ipv6
  Router ID 10.0.0.1
  Supports NSSA (compatible with RFC 3101)
  Event-log enabled, Maximum number of events: 1000, Mode: cyclic
  It is an area border and autonomous system boundary router
  Redistributing External Routes from,
     ospf 2
  Router is not originating router-LSAs with maximum metric
  Initial SPF schedule delay 5000 msecs
  Minimum hold time between two consecutive SPFs 10000 msecs
```

```
Maximum wait time between two consecutive SPFs 10000 msecs
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Retransmission limit dc 24 non-dc 24
Number of external LSA 0. Checksum Sum 0x000000
Number of areas in this router is 3. 2 normal 0 stub 1 nssa
Non-Stop Routing enabled
Graceful restart helper support enabled
Reference bandwidth unit is 100 mbps
RFC1583 compatibility enabled
   Area BACKBONE(0) (Inactive)
       Number of interfaces in this area is 2
       SPF algorithm executed 2 times
       Number of LSA 6. Checksum Sum 0x02BAB7
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
   Area 1
       Number of interfaces in this area is 4
       SPF algorithm executed 2 times
       Number of LSA 7. Checksum Sum 0x04FF3A
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
   Area 3
       Number of interfaces in this area is 1
       It is a NSSA area
       Perform type-7/type-5 LSA translation
       SPF algorithm executed 3 times
       Number of LSA 5. Checksum Sum 0x011014
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
```

The output shows that OSPFv3 NSR is configured.

Example Verifying OSPFv3 NSR

The following example shows how to verify OSPFv3 NSR status:

```
Device# show ospfv3 1 nsr
  Active RP
  Operating in duplex mode
  Redundancy state: ACTIVE
  Peer redundancy state: STANDBY HOT
  Checkpoint peer ready
  Checkpoint messages enabled
  ISSU negotiation complete
  ISSU versions compatible


          OSPFv3 1 address-family ipv4 (router-id 10.0.0.1)
  NSR configured
  Checkpoint message sequence number: 29
  Standby synchronization state: synchronized
  Bulk sync operations:  1
  Next sync check time:  12:00:14.956 PDT Wed Jun 6 2012
```

```
LSA Count: 17, Checksum Sum 0x00085289


         OSPFv3 1 address-family ipv6 (router-id 10.0.0.1)
NSR configured
Checkpoint message sequence number: 32
Standby synchronization state: synchronized
Bulk sync operations:  1
Next sync check time:  12:00:48.537 PDT Wed Jun 6 2012
LSA Count: 18, Checksum Sum 0x0008CA05
```

The output shows that OSPFv3 NSR is configured and that OSPFv3 on the standby RP is fully synchronized and ready to continue operation if the active RP fails or if a manual switchover is performed.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
OSPFv3 Address Families	*OSPFv3 Address Families* module

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 5187.	*OSPFv3 Graceful Restart*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 NSR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 282: Feature Information for OSPFv3 NSR

Feature Name	Releases	Feature Information
OSPFv3 NSR	15.1(2)SY 15.2(4)S	The OSPFv3 NSR feature allows a router with redundant RPs to maintain its OSPFv3 state and adjacencies across planned and unplanned RP switchovers. The following commands were introduced or modified: **clear ospfv3 nsr**, **nsr (OSPFv3)**, **show ospfv3 nsr**.

Table 283: Feature Information for OSPFv3 NSR

Feature Name	Releases	Feature Information
OSPFv3 NSR	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 237

OSPFv2 Loop-Free Alternate Fast Reroute

The OSPFv2 Loop-Free Alternate Fast Reroute feature uses a precomputed alternate next hop to reduce failure reaction time when the primary next hop fails. It lets you configure a per-prefix loop-free alternate (LFA) path that redirects traffic to a next hop other than the primary neighbor. The forwarding decision is made and service is restored without other routers' knowledge of the failure.

Prerequisites for OSPFv2 Loop-Free Alternate Fast Reroute

Open Shortest Path First (OSPF) supports IP FRR only on platforms that support this feature in the forwarding plane. See the Cisco Feature Navigator, http://www.cisco.com/go/cfn , for information on platform support. An account on Cisco.com is not required.

Restrictions for OSPFv2 Loop-Free Alternate Fast Reroute

The OSPFv2 Loop-Free Alternate Fast Reroute feature is not supported on routers that are virtual links headends.

The OSPFv2 Loop-Free Alternate Fast Reroute feature is supported only in global VPN routing and forwarding (VRF) OSPF instances.

You cannot configure a traffic engineering (TE) tunnel interface as a protected interface. Use the MPLS Traffic Engineering--Fast Reroute Link and Node Protection feature to protect these tunnels. See the "MPLS Traffic Engineering--Fast Reroute Link and Node Protection" section in the *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* for more information.

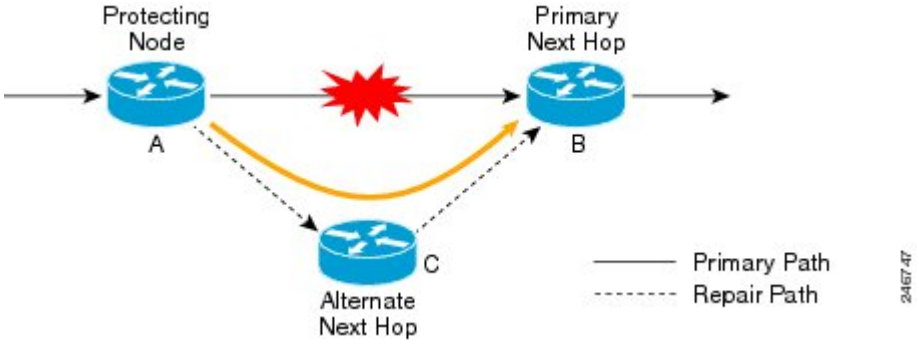
You can configure a TE tunnel interface in a repair path, but OSPF will not verify the tunnel's placement; you must ensure that it is not crossing the physical interface it is intended to protect.

Not all routes can have repair paths. Multipath primary routes might have repair paths for all, some, or no primary paths, depending on network topology, the connectivity of the computing router, and the attributes required of repair paths.

Information About OSPFv2 Loop-Free Alternate Fast Reroute

LFA Repair Paths

The figure below shows how the OSPFv2 Loop-Free Alternate Fast Reroute feature reroutes traffic if a link fails. A protecting router precomputes per-prefix repair paths and installs them in the global Routing Information Base (RIB). When the protected primary path fails, the protecting router diverts live traffic from the primary path to the stored repair path, without other routers' having to recompute network topology or even be aware that the network topology has changed.



LFA Repair Path Attributes

When a primary path fails, many paths are possible repair candidates. The OSPFv2 Loop-Free Alternate Fast Reroute feature default selection policy prioritizes attributes in the following order:

1. srlg
2. primary-path
3. interface-disjoint
4. lowest-metric
5. linecard-disjoint
6. node-protecting
7. broadcast-interface-disjoint

If the evaluation does not select any candidate, the repair path is selected by implicit load balancing. This means that repair path selection varies depending on prefix.

You can use the **show ip ospf fast-reroute** command to display the current configuration.

You can use the **fast-reroute tie-break** command to configure one or more of the repair-path attributes described in the following sections to select among the candidates:

Shared Risk Link Groups

A shared risk link group (SRLG) is a group of next-hop interfaces of repair and protected primary paths that have a high likelihood of failing simultaneously. The OSPFv2 Loop-Free Alternate Fast Reroute feature supports only SRLGs that are locally configured on the computing router. VLANs on a single physical interface are an example of an SRLG. If the physical interface fails, all the VLAN interfaces will fail at the same time. The default repair-path attributes might result in the primary path on one VLAN being protected by a repair path over another VLAN. You can configure the srlg attribute to specify that LFA repair paths do not share the same SRLG ID as the primary path. Use the **srlg** command to assign an interface to an SRLG.

Interface Protection

Point-to-point interfaces have no alternate next hop for rerouting if the primary gateway fails. You can set the interface-disjoint attribute to prevent selection of such repair paths, thus protecting the interface.

Broadcast Interface Protection

LFA repair paths protect links when a repair path and a protected primary path use different next-hop interfaces. However, on broadcast interfaces, if the LFA repair path is computed via the same interface as the primary path, but their next-hop gateways are different, the node is protected but the link might not be. You can set the broadcast-interface-disjoint attribute to specify that the repair path never crosses the broadcast network the primary path points to; that is, it cannot use the interface and the broadcast network connected to it.

See " Broadcast and Non-Broadcast Multi-Access (NBMA) Links " in RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates* for information on network topologies that require this tiebreaker.

Node Protection

The default repair-path attributes might not protect the router that is the next hop in a primary path. You can configure the node-protecting attribute to specify that the repair path will bypass the primary-path gateway router.

Downstream Path

In the case of a high-level network failure or multiple simultaneous network failures, traffic sent over an alternate path might loop until OSPF recomputes the primary paths. You can configure the downstream attribute to specify that the metric of any repair path to the protected destination must be lower than that of the protecting node to the destination. This might result in lost traffic but it prevents looping.

Line-Card Disjoint Interfaces

Line-card interfaces are similar to SRLGs because all interfaces on the same line card will fail at the same time if there is a problem with the line card, for example, line card online insertion and removal (OIR). You can configure the linecard-disjoint attribute to specify that LFA repair paths use different interfaces than those on the primary-path line card.

Metric

An LFA repair path need not be the most efficient of the candidates. A high-cost repair path might be considered more attractive if it provides protection against higher-level network failures. You can configure the metric attribute to specify a repair-path policy that has the lowest metric.

Equal-Cost Multipath Primary Paths

Equal-cost multipath paths (ECMPs) found during the primary shortest path first (SPF) repair, might not be desirable in network designs where traffic is known to exceed the capacity of any single link. You can configure the primary-path attribute to specify an LFA repair path from the ECMP set, or the secondary-path attribute to specify an LFA repair path that is not from the ECMP set.

Candidate Repair-Path Lists

When OSPF computes a repair path, it keeps in the local RIB only the best from among all the candidate paths, in order to conserve memory. You can use the **fast-reroute keep-all-paths** command to create a list of all the candidate repair paths that were considered. This information can be useful for troubleshooting but it can greatly increase memory consumption so it should be reserved for testing and debugging.

How to Configure OSPFv2 Loop-Free Alternate Fast Reroute

Enabling Per-Prefix OSPFv2 Loop-Free Alternate Fast Reroute

Perform this task to enable per-prefix OSPFv2 Loop-Free Alternate Fast Reroute and select the prefix priority in an OSPF area.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **fast-reroute per-prefix enable prefix-priority** *priority-level*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode.

	Command or Action	Purpose
Step 4	**fast-reroute per-prefix enable prefix-priority** *priority-level* **Example:** `Router (config-router)# fast-reroute per-prefix enable prefix-priority low`	Enables repair-path computation and selects the priority level for repair paths. • Low priority specifies that all prefixes have the same eligibility for protection. High priority specifies that only high-priority prefixes are protected.
Step 5	**exit** **Example:** `Router (config-router)# exit`	Exits router configuration mode and returns to global configuration mode.

Specifying Prefixes to Be Protected by LFA FRR

Perform this task to specify which prefixes will be protected by LFA FRR. Only prefixes specified in the route map will be protected.

Note Only the following three match keywords are recognized in the route map: **match tag**, **match route-type**, and **match ip address prefix-list**.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [permit | deny] [*sequence-number*]
4. **match tag** *tag-name*
5. **exit**
6. **router ospf** *process-id*
7. **prefix-priority** *priority-level* **route-map** *map-tag*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**route-map** *map-tag* [permit | deny] [*sequence-number*] **Example:** `Router(config)# route-map OSPF-PREFIX-PRIORITY`	Enters route-map configuration mode and specifies the map name.
Step 4	**match tag** *tag-name* **Example:** `Router(config-route-map)# match tag 886`	Specifies the prefixes to be matched. • Only prefixes that match the tag will be protected.
Step 5	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 6	**router ospf** *process-id* **Example:** `Router(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode.
Step 7	**prefix-priority** *priority-level* **route-map** *map-tag* **Example:** `Router(config-router)# prefix-priority high route-map OSPF-PREFIX-PRIORITY`	Sets the priority level for repair paths and specifies the route map that defines the prefixes.
Step 8	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.

Configuring a Repair Path Selection Policy

Perform this task to configure a repair path selection policy, specifying a tiebreaking condition. See the LFA Repair Path Attributes, on page 2894 for information on tiebreaking attributes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **fast-reroute per-prefix tie-break** *attribute* **[required] index** *index-level*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode.
Step 4	**fast-reroute per-prefix tie-break** *attribute* **[required] index** *index-level* **Example:** `Router(config-router)# fast-reroute per-prefix tie-break srlg required index 10`	Configures a repair path selection policy by specifying a tiebreaking condition and setting its priority level.
Step 5	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.

Creating a List of Repair Paths Considered

Perform this task to create a list of paths considered for LFA FRR.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **fast-reroute keep-all-paths**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Router(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode.
Step 4	**fast-reroute keep-all-paths** **Example:** `Router(config-router)# fast-reroute keep-all-paths`	Specifies creating a list of repair paths considered for LFA FRR.
Step 5	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.

Prohibiting an Interface From Being Used as the Next Hop

Perform this task to prohibit an interface from being used as the next hop in a repair path.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf fast-reroute per-prefix candidate disable**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Enters interface configuration mode for the interface specified.
Step 4	**ip ospf fast-reroute per-prefix candidate disable** **Example:** `Router(config-if)# ip ospf fast-reroute per-prefix candidate disable`	Prohibits the interface from being used as the next hop in a repair path.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuration Examples for OSPFv2 Loop-Free Alternate Fast Reroute

Example Enabling Per-Prefix LFA IP FRR

The following example shows how to enable per-prefix OSPFv2 Loop-Free Alternate Fast Reroute and select the prefix priority in an OSPF area:

```
Router(config)# router ospf 10
fast-reroute per-prefix enable prefix-priority low
```

Example Specifying Prefix-Protection Priority

The following example shows how to specify which prefixes will be protected by LFA FRR:

```
Router(config)# router ospf 10
prefix-priority high route-map OSPF-PREFIX-PRIORITY
fast-reroute per-prefix enable prefix-priority high
network 192.0.2.1 255.255.255.0 area 0
route-map OSPF-PREFIX-PRIORITY permit 10
match tag 866
```

Example Configuring Repair-Path Selection Policy

The following example shows how to configure a repair-path selection policy that sets SRLG, line card failure and downstream as tiebreaking attributes, and sets their priority indexes:

```
router ospf 10
  fast-reroute per-prefix enable prefix-priority low
  fast-reroute per-prefix tie-break srlg required index 10
  fast-reroute per-prefix tie-break linecard-disjoint index 15
  fast-reroute per-prefix tie-break downstream index 20
  network 192.0.2.1 255.255.255.0 area 0
```

Example Auditing Repair-Path Selection

The following example shows how to keep a record of repair-path selection:

```
router ospf 10
  fast-reroute per-prefix enable prefix-priority low
  fast-reroute keep-all-paths
  network 192.0.2.1 255.255.255.0 area 0
```

Example Prohibiting an Interface from Being a Protecting Interface

The following example shows how to prohibit an interface from being a protecting interface:

```
Router(config)# interface GigabitEthernet 0/0/0
  ip addres
s 192.0.2.1 255.255.255.0
  ip ospf fast-reroute per-prefix candidate disable
```

Additional References

The following sections provide references related to the OSPF RFC 3623 Graceful Restart feature.

Related Documents

Related Topic	Document Title
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
OSPF configuration	Configuring OSPF
Cisco nonstop forwarding	Cisco Nonstop Forwarding
OSPFv3 Graceful Restart	'*OSPFv3 Graceful Restart*' module

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2328	*OSPF Version 2*
RFC 3623	*Graceful OSPF Restart*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

Feature Information for OSPFv2 Loop-Free Alternate Fast Reroute

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 284: Feature Information for OSPFv2 Loop-Free Alternate Fast Reroute

Feature Name	Releases	Feature Information
OSPFv2 Loop-Free Alternate Fast Reroute	Cisco IOS XE Release 3.4S	This feature uses a precomputed alternate next hop to reduce failure reaction time when the primary next hop fails. The following commands were introduced or modified: **debug ip ospf fast-reroute, fast-reroute keep-all-paths, fast-reroute per-prefix (OSPF), fast-reroute tie-break (OSPF), ip ospf fast-reroute per-prefix, prefix-priority, show ip ospf fast-reroute**, **show ip ospf interface**, **show ip ospf neighbor**, **show ip ospf rib .**

Table 285: Feature Information for OSPFv2 Loop-Free Alternate Fast Reroute

Feature Name	Releases	Feature Information
OSPFv2 Loop-Free Alternate Fast Reroute	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 238

OSPFv3 MIB

The OSPFv3 MIB feature enables remote monitoring and troubleshooting of Open Shortest Path First version 3 (OSPFv3) processes using standard Simple Network Management Protocol (SNMP) management workstations. The protocol information collected by the OSPFv3 MIB objects and trap objects can be used to derive statistics that helps monitor and improve overall network performance.

Prerequisites for OSPFv3 MIB

- Ensure that Open Shortest Path First version 3 (OSPFv3) is configured on the device.
- Ensure that Simple Network Management Protocol (SNMP) is enabled on the device before notifications (traps) can be configured or before SNMP GET operations can be performed.

Restrictions for OSPFv3 MIB Support

- To monitor multiple Open Shortest Path First version 3 (OSPFv3) processes, each process must be associated with a Simple Network Management Protocol (SNMP) context.
- To monitor multiple VRFs, each VRF must be associated with an SNMP context.

Information About OSPFv3 MIB

OSPFv3 MIB

Open Shortest Path First version 3 (OSPFv3) is the IPv6 implementation of OSPF. The OSPFv3 MIB is documented in RFC 5643 and defines a MIB for managing OSPFv3 processes through Simple Network Management Protocol (SNMP).

Users can constantly monitor the changing state of an OSPF network by using MIB objects. The MIB objects gather information relating to protocol parameters and trap notification objects that can signal the occurrence of significant network events such as transition state changes.

OSPFv3 TRAP MIB

The ospfv3Notifications MIB object contains the OSPFv3 trap MIB objects that enable and disable OSPF traps in the Cisco IOS CLI. These OSPFv3 trap MIB objects are provided by the RFC 5643 standard OSPFv3 MIB.

How to Configure OSPFv3 MIB

Enabling Specific OSPFv3 Traps

Note On a Cisco Catalyst 6880-X switch, you can configure the **snmp-server enable traps ospfv3** command only with an Advanced Enterprise Services license. A Cisco Catalyst 6880-X switch operating with an IP Services license does not support the **snmp-server enable traps ospfv3** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
4. **snmp-server enable traps ospfv3 errors** [**bad-packet**] [**config-error**] [**virt-bad-packet**] [**virt-config-error**]
5. **snmp-server enable traps ospfv3 rate-limit** *seconds trap-number*
6. **snmp-server enable traps ospfv3 state-change** [**if-state-change**] [**neighbor-restart-helper-status-change**] [**neighbor-state-change**] [**nssa-translator-status-change**] [**restart-status-change**] [**virtif-state-change**] [**virtneighbor-restart-helper-status-change**] [**virtneighbor-state-change**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-serverhost** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Device(config)# snmp-server host 172.20.2.162 version 2c public ospfv3`	Specifies a recipient (target host) for Simple Network Management Protocol (SNMP) notification operations. • If the *notification-type* is not specified, all enabled notifications (traps or informs) are sent to the specified host. • If you want to send only the Open Shortest Path First version 3 (OSPFv3) notifications to the specified host, you can use the optional **ospfv3** keyword as the *notification-types* . Entering the **ospfv3** keyword enables the ospfv3Notifications MIB object.
Step 4	**snmp-server enable traps ospfv3 errors** [**bad-packet**] [**config-error**] [**virt-bad-packet**] [**virt-config-error**] **Example:** `Device(config)# snmp-server enable traps ospfv3 errors`	Enables SNMP notifications for OSPFv3 errors.
Step 5	**snmp-server enable traps ospfv3 rate-limit** *seconds trap-number* **Example:** `Device(config)# snmp-server enable traps ospfv3 rate-limit 20 20`	Sets the rate limit for the number of SNMP OSPFv3 notifications that are sent in each OSPFv3 SNMP notification rate-limit window.
Step 6	**snmp-server enable traps ospfv3 state-change** [**if-state-change**] [**neighbor-restart-helper-status-change**] [**neighbor-state-change**] [**nssa-translator-status-change**] [**restart-status-change**] [**virtif-state-change**] [**virtneighbor-restart-helper-status-change**] [**virtneighbor-state-change**] **Example:** `Device(config)# snmp-server enable traps ospfv3 state-change`	Enables SNMP OSPFv3 notifications for OSPFv3 transition state changes.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

Verifying OSPFv3 MIB Traps on the Device

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show running-config** [*options*]

Example:

```
Device# show running-config | include traps
```

Displays the contents of the currently running configuration file and includes information about enabled traps.

- Verifies which traps are enabled.

Configuration Examples for OSPFv3 MIB

Example: Enabling and Verifying OSPFv3 MIB Traps

The following example shows how to enable all OSPFv3 error traps:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server enable traps ospfv3 errors
Device(config)# end
```

The following example shows how to verify that the traps are enabled:

```
Device> enable
Device# show running-config | include traps

snmp-server enable traps ospfv3 errors
```

Additional References for OSPFv3 MIB

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
OSPF configuration tasks	"Configuring OSPF" module in *IP Routing: OSPF Configuration Guide*

Standards and RFCs

Standard	Title
RFC 5643	*Management Information Base for OSPFv3*

MIBs

MIB	MIBs Link
OSPFv3-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 286: Feature Information for OSPFv3 MIB

Feature Name	Releases	Feature Information
OSPFv3 MIB	Cisco IOS XE Release 3.7S	The OSPFv3 MIB feature enables remote monitoring and troubleshooting of OSPFv3 processes using standard SNMP management workstations. The following commands were introduced or modified: **snmp-server host**, **snmp-server enable traps ospfv3 errors**, **snmp-server enable traps ospfv3 rate-limit**, **snmp-server enable traps ospfv3 state-change**.

Table 287: Feature Information for OSPFv3 MIB

Feature Name	Releases	Feature Information
OSPFv3 MIB	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 239

Prefix Suppression Support for OSPFv3

This feature enables Open Shortest Path First version 3 (OSPFv3) to hide the IPv4 and IPv6 prefixes of connected networks from link-state advertisements (LSAs). When OSPFv3 is deployed in large networks, limiting the number of IPv4 and IPv6 prefixes that are carried in the OSPFv3 LSAs can speed up OSPFv3 convergence.

This feature can also be utilized to enhance the security of an OSPFv3 network by allowing the network administrator to prevent IP routing toward internal nodes.

Prerequisites for Prefix Suppression Support for OSPFv3

Before you can use the mechanism to exclude IPv4 and IPv6 prefixes from LSAs, the OSPFv3 routing protocol must be configured.

Information About Prefix Suppression Support for OSPFv3

OSPFv3 Prefix Suppression Support

The OSPFv3 Prefix Suppression Support feature allows you to hide IPv4 and IPv6 prefixes that are configured on interfaces running OSPFv3.

In OSPFv3, addressing semantics have been removed from the OSPF protocol packets and the main LSA types, leaving a network-protocol-independent core. This means that Router-LSAs and network-LSAs no longer contain network addresses, but simply express topology information. The process of hiding prefixes is simpler in OSPFv3 and suppressed prefixes are simply removed from the intra-area-prefix-LSA. Prefixes are also propagated in OSPFv3 via link LSAs

The OSPFv3 Prefix Suppression feature provides a number of benefits.The exclusion of certain prefixes from adverstisements means that there is more memory available for LSA storage, bandwidth and buffers for LSA

flooding, and CPU cycles for origination and flooding of LSAs and for SPF computation. Prefixes are also filtered from link LSAs. A device only filters locally configured prefixes, not prefixes learnt via link LSAs. In addition, security has been improved by reducing the possiblity of remote attack with the hiding of transit-only networks.

Globally Suppress IPv4 and IPv6 Prefix Advertisements by Configuring the OSPFv3 Process

You can reduce OSPFv3 convergence time by configuring the OSPFv3 process on a device to prevent the advertisement of all IPv4 and IPv6 prefixes by using the **prefix-suppression** command in router configuration mode or address-family configuration mode.

Note Prefixes that are associated with loopbacks, secondary IP addresses, and passive interfaces are not suppressed by the **router mode** or the **address-family** configuration commands because typical network designs require prefixes to remain reachable.

Suppress IPv4 and IPv6 Prefix Advertisements on a Per-Interface Basis

You can explicitly configure an OSPFv3 interface not to advertise its IP network to its neighbors by using the **ipv6 ospf prefix-suppression** command or the **ospfv3 prefix-suppression** command in interface configuration mode.

Note If you have globally suppressed IPv4 and IPv6 prefixes from connected IP networks by configuring the **prefix-suppression** router configuration command, the interface configuration command takes precedence over the router configuration command.

How to Configure Prefix Suppression Support for OSPFv3

Configuring Prefix Suppression Support of the OSPFv3 Process

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** *process-id* [**vrf** *vpn-name*]
4. **prefix-suppression**
5. **end**
6. **show ospfv3**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# router ospfv3 23`	Configures an OSPFv3 routing process and enters router configuration mode.
Step 4	**prefix-suppression** **Example:** `Device(config-router)# prefix-suppression`	Prevents OSPFv3 from advertising all IPv4 and IPv6 prefixes, except prefixes that are associated with loopbacks, secondary IP addresses, and passive interfaces.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to privileged EXEC mode.
Step 6	**show ospfv3** **Example:** `Device# show ospfv3`	Displays general information about OSPFv3 routing processes. **Note** Use this command to verify that IPv4 and IPv6 prefix suppression has been enabled.

Configuring Prefix Suppression Support of the OSPFv3 Process in Address-Family Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** *process-id* [**vrf** *vpn-name*]
4. **address-family ipv6 unicast**
5. **prefix-suppression**
6. **end**
7. **show ospfv3**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** *process-id* [**vrf** *vpn-name*] **Example:** `Device(config)# router ospfv3 23`	Configures an OSPFv3 routing process and enters router configuration mode.
Step 4	**address-family ipv6 unicast** **Example:** `Device(config-router)# address-family ipv6 unicast`	Enters IPv6 address family configuration mode for OSPFv3.
Step 5	**prefix-suppression** **Example:** `Device(config-router-af)# prefix-suppression`	Prevents OSPFv3 from advertising all IPv4 and IPv6 prefixes, except prefixes that are associated with loopbacks, secondary IP addresses, and passive interfaces.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Returns to privileged EXEC mode.
Step 7	**show ospfv3** **Example:** `Device# show ospfv3`	Displays general information about OSPFv3 routing processes. **Note** Use this command to verify that IPv4 and IPv6 prefix suppression has been enabled.

Configuring Prefix Suppression Support on a Per-Interface Basis

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ipv6 ospf prefix-suppression** [**disable**]

- **ospfv3 prefix-suppression disable**

5. **end**
6. **show ospfv3 interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial 0/0`	Configures an interface type and enters interface configuration mode.

Step 4	Do one of the following:
	• **ipv6 ospf prefix-suppression** [**disable**]
	• **ospfv3 prefix-suppression disable**
	Example:
	`Device(config-if)# ipv6 ospf prefix-suppression`
	Example:
	`Device(config-if)# ospfv3 1 prefix-suppression disable`
		Prevents OSPFv3 from advertising IPv4 and IPv6 prefixes that belong to a specific interface, except those that are associated with secondary IP addresses.
		• When you enter the **ipv6 ospf prefix-suppression** command or the **ospfv3 prefix-suppression** command in interface configuration mode, it takes precedence over the **prefix-suppression** command that is entered in router configuration mode.

Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ospfv3 interface** **Example:** `Device# show ospfv3 interface`	Displays OSPFv3-related interface information. **Note** Use this command to verify that IPv4 and IPv6 prefix suppression has been enabled for a specific interface.

Troubleshooting IPv4 and IPv6 Prefix Suppression

SUMMARY STEPS

1. **enable**
2. **debug ospfv3 lsa-generation**
3. **debug condition interface** *interface-type interface-number* [**dlci** *dlci*] [**vc** {*vci* | *vpi* | *vci*}]
4. **show debugging**
5. **show logging** [**slot** *slot-number* | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug ospfv3 lsa-generation** **Example:** `Device# debug ospfv3 lsa-generation`	Displays informations about each OSPFv3 LSA that is generated.
Step 3	**debug condition interface** *interface-type interface-number* [**dlci** *dlci*] [**vc** {*vci* | *vpi* | *vci*}] **Example:** `Device# debug condition interface serial 0/0`	Limits output for some **debug** commands on the basis of the interface or virtual circuit.
Step 4	**show debugging** **Example:** `Device# show debugging`	Displays information about the types of debugging that are enabled for your device.
Step 5	**show logging** [**slot** *slot-number* | **summary**] **Example:** `Device# show logging`	Displays the state of syslog and the contents of the standard system logging buffer.

Configuration Examples for Prefix Suppression Support for OSPFv3

Example: Configuring Prefix Suppression Support for OSPFv3

The following example shows how to configure prefix suppression support for OSPFv3 in router configuration mode:

```
router ospfv3 1
 prefix-suppression
 !
 address-family ipv6 unicast
  router-id 0.0.0.6
  exit-address-family
```

The following example shows how to configure prefix suppression support for OSPFv3 in address-family configuration mode:

```
router ospfv3 1
 !
 address-family ipv6 unicast
  router-id 10.0.0.6
  prefix-suppression
  exit-address-family
```

The following example shows how to configure prefix suppression support for OSPFv3 in interface configuration mode:

```
interface Ethernet0/0
 ip address 10.0.0.1 255.255.255.0
 ipv6 address 2001:201::201/64
 ipv6 enable
 ospfv3 prefix-suppression
 ospfv3 1 ipv4 area 0
 ospfv3 1 ipv6 area 0
 end
```

Additional References for Prefix Suppression Support for OSPFv3

Related Documents

Related Topic	Document Title
Configuring OSPF	“Configuring OSPF”
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Prefix Suppression Support for OSPFv3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 288: Feature Information for Prefix Suppression Support for OSPFv3

Feature Name	Releases	Feature Information
Prefix Suppression Support for OSPFv3	Cisco IOS XE Release 3.8S	This feature enables Open Shortest Path First version 3 (OSPFv3) to hide the IPv4 and IPv6 prefixes of connected networks from link-state advertisements (LSAs). This feature can also be used to enhance the security of an OSPFv3 network by allowing the network administrator to prevent IP routing toward internal nodes. The following commands were introduced or modified: • **ipv6 ospf prefix-suppression** • **ospfv3 prefix-suppression** • **prefix-suppression (OSPFv3)**

Table 289: Feature Information for Prefix Suppression Support for OSPFv3

Feature Name	Releases	Feature Information
Prefix Suppression Support for OSPFv3	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 240

OSPFv3 VRF-Lite/PE-CE

The OSPFv3 VRF-Lite/PE-CE feature adds Open Shortest Path First version 3 (OSPFv3) support for nondefault VPN routing and forwarding (VRF) instances. OSPFv3 can be used as a provider-edge-customer-edge (PE-CE) routing protocol as specified in RFC 6565, *OSPFv3 as a Provider Edge to Customer Edge (PE-CE) Routing Protocol*. OSPFv3 in a nondefault VRF instance supports routing of IPv4 and IPv6 address families.

Restrictions for OSPFv3 VRF-Lite/PE-CE

In Cisco IOS Release 15.2(2)S and later releases, OSPFv3 interface commands in the **ipv6 ospf** format are no longer supported in VRF interface configuration mode. You must configure them in the new format, **ospfv3**.

The **ospfv3** commands can have one of following formats:

- **ospfv3** —Applies to all OSPFv3 processes and address families on a given interface.
- **ospfv3** *process-id* —Applies to an OSPFv3 process with the configured process ID and to both IPv4 and IPv6 address families.
- **ospfv3** *process-id address-family-ID* —Applies to an OSPFv3 process with the configured process ID and the configured address family.

More specific commands take precedence over less specific commands, as shown in the following descending order:

1. Commands that specify a process ID and an address family.
2. Commands that specify only a process ID.
3. Commands that specify neither a process ID nor an address family.

In Cisco IOS Release 15.2(2)S and later releases, you cannot use the **ipv6 ospf router** *process-id* command to configure OSPFv3 VRF instances. You must configure the **router ospfv3** *process-id* command in global configuration mode and specify the address family for the configured VRF in router configuration mode.

Information About OSPFv3 VRF-Lite/PE-CE

Support for OSPFv3 VRF-Lite and PE-CE

Open Shortest Path First version 3 (OSPFv3) operates in nondefault VPN routing and forwarding (VRF) instances for both IPv6 and IPv4 address families and, transports the routes across a Border Gateway Protocol (BGP) or a Multiprotocol Label Switching (MPLS) backbone. On the provider edge (PE) device, customer routes are installed together by OSPFv3 and BGP in a common VRF or address family and each protocol is configured to redistribute the routes of the other. BGP combines the prefixes redistributed into it with a route-distinguisher value defined for the VRF and advertises them to other MPLS-BGP speakers in the same autonomous system using the VPNv4 or VPNv6 address family as appropriate.

The OSPFv3 route selection algorithm prefers intra-area routes across the back-door link over inter-area routes through the MPLS backbone. Sham-links are a type of virtual link across the MPLS backbone that connect OSPFv3 instances on different PEs. OSPFv3 instances tunnel protocol packets through the backbone and form adjacencies. Because OSPFv3 considers the sham-link as an intra-area connection, sham-link serves as a valid alternative to an intra-area back-door link.

Domain IDs are used to determine whether the routes are internal or external. They describe the administrative domain of the OSPFv3 instance from which the route originates. Every PE has a 48-bit primary domain ID (which may be NULL) and zero or more secondary domain IDs.

How to Configure VRF-Lite/PE-CE

Configuring a VRF in an IPv6 Address Family for OSPFv3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **router ospfv3** [*process-id*]
7. **address-family ipv6** [**unicast**] [**vrf** *vrf-name*]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrfsample`	Configures a VRF routing table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 10:1`	Creates routing and forwarding tables for a VRF.
Step 5	**exit** **Example:** `Device(config-vrf)# exit`	Exists VRF configuration mode and returns to global configuration mode.
Step 6	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 2`	Configures an OSPF routing process and enters router configuration mode.
Step 7	**address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv6 unicast vrf vrfsample`	Configures an instance of the OSPFv3 process in the VRF routing table for the IPv6 address family and enters router address family configuration mode.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exists router address family configuration mode and returns to privileged EXEC mode.

Enabling an OSPFv3 IPv6 Address Family on a VRF Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name* [**downstream** *vrf-name2*]
5. **ipv6 enable**
6. **ospfv3** *process-id* {**ipv4** | **ipv6**} **area** *area-id* [**instance** *instance-id*]

7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Serial6/0`	Specifies an interface type and number and enters interface configuration mode.
Step 4	**vrf forwarding** *vrf-name* [**downstream** *vrf-name2*] **Example:** `Device(config-if)# vrf forwarding v1`	Associates an interface with a VRF.
Step 5	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on the interface that is associated with the VRF.
Step 6	**ospfv3** *process-id* {**ipv4** | **ipv6**} **area** *area-id* [**instance** *instance-id*] **Example:** `Device(config-if)# ospfv3 1 ipv6 area 0`	Enables the OSPFv3 IPv6 address family on the VRF interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Sham-Link for OSPFv3 PE-CE

Before you begin

The OSPFv3 PE-CE feature supports direct forwarding on Border Gateway Protocol (BGP) routes.

Before you configure a sham-link, you must create a Multiprotocol Label Switching (MPLS) backbone, configure a device as an MPLS VPN PE device, and configure OSPFv3 as the provider-edge-customer-edge (PE-CE) protocol in a virtual routing and forwarding (VRF) instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface-number*
4. **description** *string*
5. **vrf forwarding** *vrf-name*
6. **ipv6 address** *ipv6-address/prefix-length*
7. **ipv6 enable**
8. **end**
9. **router ospfv3** *process-id*
10. **address-family** {**ipv4** | **ipv6**} [**unicast** | **multicast**] [**vrf** *vrf-name*]
11. **redistribute** *process-id* [*options*]
12. **area** *area-id* **sham-link** *source-address destination-address* [**cost** *number*] [**ttl-security hops** *hop-count*]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface-number* **Example:** `Device(config)# interface loopback 0`	Creates a loopback interface to be used as an endpoint of the sham-link on a provider edge device and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-if)# description Sham-link endpoint`	Provides a description of the interface to help you track its status.
Step 5	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates the loopback interface with a VRF.

	Command or Action	Purpose
Step 6	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:0:ABCD::1/48`	Configures an IPv6 address of the loopback interface on a provider edge device.
Step 7	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on the loopback interface.
Step 8	**end** **Example:** `Device# end`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**router ospfv3** *process-id* **Example:** `Device(config)# router ospfv3 1`	Enters router configuration mode.
Step 10	**address-family** {**ipv4** | **ipv6**} [**unicast** | **multicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv6 unicast vrf vrf1`	Enters IPv6 address family configuration mode for OSPFv3.
Step 11	**redistribute** *process-id* [*options*] **Example:** `Device(config-router-af)# redistribute bgp 2`	Redistributes IPv6 routes from the specified source BGP routing domain into the specified destination routing domain. **Note** PE-CE redistribution is always from BGP.
Step 12	**area** *area-id* **sham-link** *source-address destination-address* [**cost** *number*] [**ttl-security hops** *hop-count*] **Example:** `Device(config-router-af)# area 0 sham-link 2001:DB8:0:ABCD::1 2001:DB8:0:ABCD::2 cost 100`	Enables the sham-link and specifies its source and destination addresses.
Step 13	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring a Domain ID for an OSPFv3 PE-CE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **router ospfv3** [*process-id*]
7. **address-family ipv6** [**unicast**] [**vrf** *vrf-name*]
8. **domain-id type** *type* **value** *hex-value*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrfsample`	Configures a VRF routing table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 10:1`	Creates routing and forwarding tables for a VRF.
Step 5	**exit** **Example:** `Device(config-vrf)# exit`	Exists VRF configuration mode and returns to global configuration mode.
Step 6	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 2`	Enters router configuration mode.

	Command or Action	Purpose
Step 7	**address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv6 unicast vrf vrfsample`	Configures an instance of the OSPFv3 process in the VRF routing table for the IPv6 address family and enters address family configuration mode..
Step 8	**domain-id type** *type* **value** *hex-value* **Example:** `Device(config-router-af)# domain-id type 0205 value 800EFFFF12AB`	Configures the BGP domain ID. • The value for **type** can be 0005, 0105, 0205, or 8005. • The value for **value** is an arbitrary 48-bit number encoded as 12 hexadecimal digits.
Step 9	**end** **Example:** `Device(config-router-af)# end`	Exists router address family mode and returns to privileged EXEC mode.

Configuring VRF-Lite Capability for OSPFv3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **router ospfv3** [*process-id*]
7. **address-family ipv6** [**unicast**] [**vrf** *vrf-name*]
8. **capability vrf-lite**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrfsample`	Configures a VRF routing table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 10:1`	Creates routing and forwarding tables for a VRF.
Step 5	**exit** **Example:** `Device(config-vrf)# exit`	Exists VRF configuration mode and returns to global configuration mode.
Step 6	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 2`	Enables router configuration mode for the IPv4 or IPv6 address family.
Step 7	**address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **Example:** `Device(config-router)# address-family ipv6 unicast vrf vrfsample`	Configures an instance of the OSPFv3 process in the VRF routing table for the IPv6 address family and enters address family configuration mode.
Step 8	**capability vrf-lite** **Example:** `Device(config-router-af)# capability vrf-lite`	Applies the multi-VRF capability to the OSPF process.
Step 9	**end** **Example:** `Device(config-router-af)# end`	Exists router address family mode and returns to privileged EXEC mode.

Configuration Examples for OSPFv3 VRF-Lite/PE-CE

Example: Configuring a Provider Edge Device to Provide IPv6 and IPv4 Routing

The following example shows how to configure a provider edge (PE) device to provide IPv6 and IPv4 routing for a user on VRF “v1” and IPv6 routing for a user on VRF “v2”:

```
vrf definition v1
```

```
 rd 1:1
 route-target export 100:1
 route-target import 100:1
!
address-family ipv4
 exit-address-family
!
address-family ipv6
 exit-address-family
!
vrf definition v2
 rd 2:2
 route-target export 200:2
 route-target import 200:2
!
address-family ipv6
 exit-address-family
!
interface Loopback1
 vrf forwarding v1
 ipv6 address 2001:DB8:0:ABCD::1/48
!
interface Serial5/0
 vrf forwarding v2
 no ip address
 ipv6 address 2001:DB8:0:ABCD::3/48
 ospfv3 1 ipv6 area 1
!
interface Serial6/0
 vrf forwarding v1
 ip address 10.0.0.1 255.255.255.0
 ipv6 enable
 ospfv3 1 ipv6 area 0
 ospfv3 1 ipv4 area 10.1.1.1
!
router ospfv3
!
log-adjacency-changes detail
!
address-family ipv4 unicast vrf v1
 router-id 10.2.2.2
 redistribute bgp 1
 exit-address-family
!
address-family ipv6 unicast vrf v1
 router-id 2001:DB8:1::1
 domain-id type 0205 value 111111222222
 area 0 sham-link 2001:DB8:0:ABCD::5 2001:DB8:0:ABCD::7
 redistribute bgp 1
 exit-address-family
address-family ipv6 unicast vrf v2
 router-id 2001:DB8:1::3
 redistribute bgp 1
 exit
!
router bgp 1
 bgp router-id 10.3.3.3
 no bgp default ipv4-unicast
 neighbor 10.0.0.4 remote-as 1
 neighbor 10.0.0.4 update-source-Loopback0
!
address-family ipv4
 exit-address-family
!
```

```
address-family vpnv4
 neighbor 10.0.0.4
 neighbor 10.0.0.4 send-community extended
 exit-address-family
!
address-family vpnv6
 neighbor 10.0.0.4 activate
 neighbor 10.0.0.4 send-community extended
 exit-address-family
!
address-family ipv4 vrf v1
 redistribute ospfv3 1
 exit-address-family
!
address-family ipv6 vrf v1
 redistribute ospf 1
 exit-address-family
!
address-family ipv6 vrf v2
 redistribute ospf 1
 exit-address-family
!
```

Example: Configuring a Provider Edge Device for VRF-Lite

```
vrf definition v1
 rd 1:1
 !
address-family ipv4
 exit-address-family
!
address-family ipv6
 exit-address-family
!
vrf definition v2
 rd 2:2
 !
address-family ipv6
 exit-address-family
!
interface FastEthernet0/0
 no ip address
 !
interface FastEthernet0/0.100
 encapsulation dot1Q 100
 vrf forwarding v1
 ip address 192.168.1.1 255.255.255.0
 ipv6 enable
 ospfv3 1 ipv6 area 0
 ospfv3 1 ipv4 area 0
!
interface FastEthernet0/0.200
 encapsulation dot1Q 200
 vrf forwarding v2
 ipv6 enable
 ospfv3 1 ipv6 area 0
!
interface FastEthernet0/1
 rf forwarding v1
 ip address 10.1.1.1 255.255.255.0
```

```
 ipv6 enable
 ospfv3 1 ipv6 area 1
 ospfv3 1 ipv4 area 0
 no keepalive
!
interface FastEthernet0/2
 vrf forwarding v2
 no ip address
 ipv6 address 2001:DB8:1::1
 ipv6 enable
 ospfv3 1 ipv6 area 1
!
router ospfv3 1
!
address-family ipv6 unicast vrf v2
 router-id 192.168.2.1
 capability vrf-lite
 exit-address-family
!
address-family ipv4 unicast vrf v1
  router-id 192.168.1.4
  capability vrf-lite
exit-address-family
!
address-family ipv6 unicast vrf v1
 router-id 192.168.1.1
 capability vrf-lite
 exit-address-family
!
```

Additional References for OSPFv3 VRF-Lite/PE-CE

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference

RFCs

RFC	Title
RFC 5838	Support of Address Families in OSPFv3
RFC 6565	OSPFv3 as a Provider Edge to Customer Edge (PE-CE) Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 VRF-Lite/PE-CE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 290: Feature Information for OSPFv3 VRF-Lite/PE-CE

Feature Name	Releases	Feature Information
OSPFv3 VRF-Lite/PE-CE	Cisco IOS XE Release 3.6S	The OSPFv3 VRF-Lite/PE-CE feature adds OSPFv3 support for nondefault VRF instances. The following commands were introduced or modified: **area sham-link** (OSPFv3), **capability vrf-lite** (OSPFv3).

Table 291: Feature Information for OSPFv3 VRF-Lite/PE-CE

Feature Name	Releases	Feature Information
OSPFv3 VRF-Lite/PE-CE	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 241

OSPFv3 ABR Type 3 LSA Filtering

This feature extends the ability of an Area Border Router (ABR) that is running the Open Shortest Path First version 3 (OSPFv3) protocol to filter type 3 link-state advertisements (LSAs) that are sent between different OSPFv3 areas. This feature allows only packets with specified prefixes to be sent from one area to another area and restricts all packets with other prefixes. This type of area filtering can be applied out of a specific OSPFv3 area, into a specific OSPFv3 area, or into and out of the same OSPFv3 areas at the same time.

OSPFv3 ABR Type 3 LSA Filtering

Only type 3 LSAs that originate from an ABR are filtered.

Information About OSPFv3 ABR Type 3 LSA Filtering

Area Filter Support

OSPFv3 area filters allow the filtering of inter-area prefix LSAs on the ABRs. The filter, based on IPv6 prefix lists, can be applied in both directions. In the “in” direction, it filters out the LSAs coming from all other areas when sending the inter-area prefix LSAs into the specified area. In the “out” direction, it filters out the inter-area prefix LSAs generated for the specified area.

The Area Filter Support feature gives the administrator improved control of route distribution between OSPFv3 areas.

How to Configure OSPFv3 ABR Type 3 LSA Filtering

Configuring Area Filter Support for OSPFv3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** *process-id*
4. **area** *area-id* **filter-list prefix** *prefix-list-name* {**in** | **out**}
5. **end**
6. **ipv6 prefix-list** *list-name* [**seq** *seq-number*] {**deny** *ipv6-prefix/prefix-length* | **permit** *ipv6-prefix/prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospfv3** *process-id* **Example:** `Device(config)# router ospfv3 1`	Configures the router to run an OSPFv3 process.
Step 4	**area** *area-id* **filter-list prefix** *prefix-list-name* {**in**	**out**} **Example:** `Device(config-router)# area 1 filter-list prefix test_ipv6 out`	Configures the router to filter interarea routes out of the specified area.
Step 5	**end** **Example:** `Device(config-router)# end`	Returns to global configuration mode.
Step 6	**ipv6 prefix-list** *list-name* [**seq** *seq-number*] {**deny** *ipv6-prefix/prefix-length*	**permit** *ipv6-prefix/prefix-length*	**description** *text*} [**ge** *ge-value*] [**le** *le-value*]	Creates a prefix list with the name specified for the *list-name* argument.

	Command or Action	Purpose
	Example: `Device(config)# ipv6 prefix-list test_ipv6 seq 5 permit 2011::1/128`	

Configuration Examples for OSPFv3 ABR Type 3 LSA Filtering

Example: Area Filter Support for OSPFv3

The following example shows how to configure Area Filter Support for OSPFv3:

```
router ospfv3 1
!
address-family ipv4 unicast
 area 2 filter-list prefix test_ipv4 in
exit-address-family
!
address-family ipv6 unicast
 area 2 filter-list prefix test_ipv6 in
exit-address-family
!
ip prefix-list test_ipv4 seq 5 permit 2.2.2.2/32
!
!
ipv6 prefix-list test_ipv6 seq 5 deny 2011::1/128
```

Additional References for OSPFv3 ABR Type 3 LSA Filtering

Related Documents

Related Topic	Document Title
Configuring OSPF	"Configuring OSPF"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS master command list, all releases	*Cisco IOS Master Command List, All Releases*

Standards

Standard	Title
No new or modified standards are supported and support for existing standards has not been modified.	—

RFCs

RFC	Title
No new or modified RFCs are supported and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 ABR Type 3 LSA Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 292: Feature Information for OSPFv3 ABR Type 3 LSA Filtering

Feature Name	Releases	Feature Information
OSPFv3 ABR Type 3 LSA Filtering	Cisco IOS XE Release 3.8 15.3(1)S 15.2(1)E	The OSPFv3 ABR Type 3 LSA Filtering feature extends the ability of an ABR that is running the OSPFv3 protocol to filter type 3 LSAs that are sent between different OSPFv3 areas. This feature allows only packets with specified prefixes to be sent from one area to another area and restricts all packets with other prefixes. This type of area filtering can be applied out of a specific OSPFv3 area, into a specific OSPFv3 area, or into and out of the same OSPFv3 areas at the same time.

Table 293: Feature Information for OSPFv3 ABR Type 3 LSA Filtering

Feature Name	Releases	Feature Information
OSPFv3 ABR Type 3 LSA Filtering	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 242

OSPFv3 Demand Circuit Ignore

This feature enables you to prevent an interface from accepting demand-circuit requests from other devices by specifying the ignore keyword in the **ipv6 ospf demand-circuit** command.

Information About OSPFv3 Demand Circuit Ignore

Demand Circuit Ignore Support

Demand Circuit Ignore Support enables you to prevent an interface from accepting demand-circuit requests from other devices by specifying the ignore keyword in the **ipv6 ospf demand-circuit** command. Demand circuit ignore instructs the router not to accept Demand Circuit (DC) negotiation and is a useful configuration option on the point-to-multipoint interface of the Hub router.

How to Configure OSPFv3 Demand Circuit Ignore

Configuring Demand Circuit Ignore Support for OSPFv3

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Enter one of the following commands:
   - **ipv6 ospf demand-circuit ignore**
   - **ospfv3 demand-circuit ignore**

5. **end**
6. **show ospfv3** *process-id* [*area-id*] [*address-family*] **[vrf** {*vrf-name* **|*** }] **interface** [*type number*] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/0`	Configures an interface type and number and enters interface configuration mode.

Step 4	Enter one of the following commands:
	• **ipv6 ospf demand-circuit ignore**
	• **ospfv3 demand-circuit ignore**
	Example:
	`Device(config-if)# ipv6 ospf demand-circuit ignore`
	Example:
	`Device(config-if)# ospfv3 demand-circuit ignore`
		Prevents an interface from accepting demand-circuit requests from other devices.

Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ospfv3** *process-id* [*area-id*] [*address-family*] **[vrf** {*vrf-name* **|*** }] **interface** [*type number*] [**brief**] **Example:** `Device# show ospfv3 interface GigabitEthernet 0/1/0`	(Optional) Displays OSPFv3-related interface information.

Configuration Examples for OSPFv3 Demand Circuit Ignore

Example: Demand Circuit Ignore Support for OSPFv3

The following example shows how to configure demand circuit ignore support for OSPFv3:

```
interface Serial0/0
 ip address 6.1.1.1 255.255.255.0
 ipv6 enable
 ospfv3 network point-to-multipoint
 ospfv3 demand-circuit ignore
 ospfv3 1 ipv6 area 0
```

Additional References for OSPFv3 Demand Circuit Ignore

The following sections provide references related to the OSPFv3 Demand Circuit Ignore feature.

Related Documents

Related Topic	Document Title
OSPF configuration tasks	"Configuring OSPF"
OSPF commands	*Cisco IOS IP Routing: OSPF Command Reference*
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv3 Demand Circuit Ignore

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 294: Feature Information for OSPFv3 Demand Circuit Ignore

Feature Name	Releases	Feature Information
OSPFv3 Demand Circuit Ignore	Cisco IOS XE Release 3.8	The OSPFv3 Demand Circuit Ignore feature enables you to prevent an interface from accepting demand-circuit requests from other devices by specifying the ignore keyword in the **ipv6 ospf demand-circuit** command. The following commands were introduced or modified: • **ipv6 ospf demand-circuit** • **ospfv3 demand-circuit**

Table 295: Feature Information for OSPFv3 Demand Circuit Ignore

Feature Name	Releases	Feature Information
OSPFv3 Demand Circuit Ignore	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 243

OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

OSPF IPv4 remote loop-free alternate (LFA) IP fast reroute (IPFRR) uses a backup route, precomputed using the dynamic routing protocol, whenever a network fails. The backup routes (repair paths) are pre-computed and installed in the router as the backup for the primary paths. Once the router detects a link or adjacent node failure, it switches to the backup path to avoid traffic loss.

OSPF IPv4 remote LFA IPFRR allows the backup path to be more than one hop away. This feature is particularly useful in some topologies (such as the commonly used ring topology) where an LFA does not have to be directly connected to the protecting router.

Prerequisites for OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

- Before performing the tasks in this module, you should be familiar with the concepts described in the "OSPFv2 Loop-Free Alternate Fast Reroute" module.
- LFA must be enabled.
- Your network must be configured for Multiprotocol Label Switching (MPLS).

Restrictions for OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

- The OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute feature is not supported on devices that are virtual links headends.
- The feature is supported only in global VPN routing and forwarding (VRF) OSPF instances.
- The only supported tunneling method is MPLS.
- You cannot configure a traffic engineering (TE) tunnel interface as a protected interface. Use the MPLS Traffic Engineering—Fast Reroute Link and Node Protection feature to protect these tunnels. For more information, see the "MPLS Traffic Engineering—Fast Reroute Link and Node Protection" section in the *Multiprotocol Label Switching Configuration Guide*.
- You can configure a TE tunnel interface in a repair path, but OSPF will not verify the tunnel's placement; you must ensure that it is not crossing the physical interface that it is intended to protect.
- Not all routes can have repair paths. Multipath primary routes might have repair paths for all, some, or no primary paths, depending on the network topology, the connectivity of the computing router, and the attributes required of repair paths.
- Devices that can be selected as tunnel termination points must have a /32 address advertised in the area in which remote LFA is enabled. This address will be used as a tunnel termination IP. If the device does not advertise a /32 address, it may not be used for remote LFA tunnel termination.
- All devices in the network that can be selected as tunnel termination points must be configured to accept targeted LDP sessions using the **mpls ldp discovery targeted-hello accept** command.

Information About OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

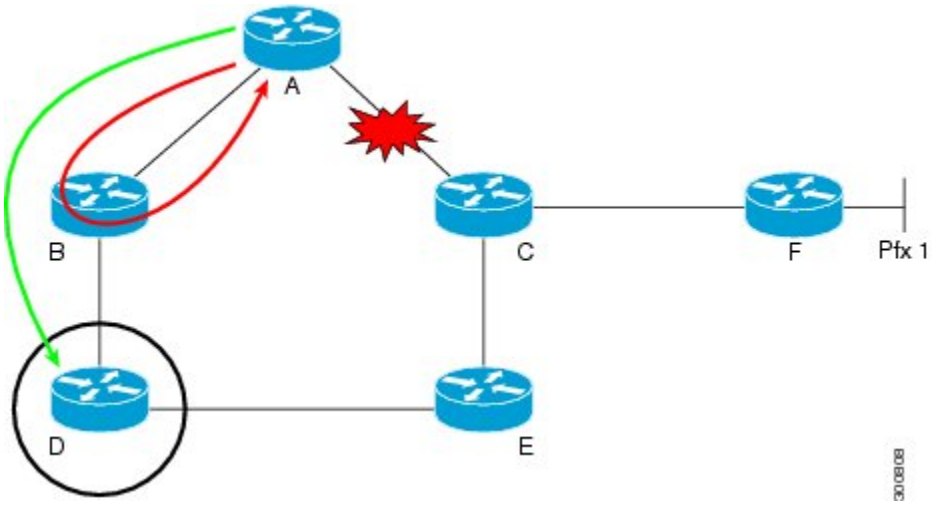
IP Fast Reroute

The IP fast reroute (IPFRR) LFA computation provides protection against link failure. Locally computed repair paths are used to prevent packet loss caused by loops that occur during network reconvergence after a failure. For more information about IPFRR, see RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates*.

OSPF IPv4 Remote LFA IPFRR with Ring Topology

Some topologies (for example the commonly used ring-based topology) require protection that is not afforded by LFA FRR alone. Consider the topology shown in the figure below:

Figure 212: Remote LFA IPFRR with Ring Topology



The red looping arrow represents traffic that is looping immediately after a failure between node A and C (before network reconvergence). Device A tries to send traffic destined to F to next-hop B. Device B cannot be used as an LFA for prefixes advertised by nodes C and F. The actual LFA is node D. However, node D is not directly connected to the protecting node A. To protect prefixes advertised by C, node A must tunnel the packet around the failed link A-C to node D, provided that the tunnel does not traverse the failing link.

The OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute feature enables you to tunnel a packet around a failed link to a remote loop-free alternate that is more than one hop away. In the figure above, the green arrow between A and D shows the tunnel that is automatically created by the OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute feature to bypass looping.

Note In the figure above, device A must be configured with **fast-reroute per-prefix remote-lfa tunnel mpls-ldp** to enable remote LFA, and device D must be configured with **mpls ldp discovery targeted-hello accept** to accept targeted LDP sessions.

How to Configure OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

Configuring a Remote LFA Tunnel

Perform this task to configure a per-prefix LFA FRR path that redirects traffic to a remote LFA tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **fast-reroute per-prefix remote-lfa** [**area** *area-id*] **tunnel mpls-ldp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode.
Step 4	**fast-reroute per-prefix remote-lfa** [**area** *area-id*] **tunnel mpls-ldp** **Example:** `Device(config-router)# fast-reroute per-prefix remote-lfa area 2 tunnel mpls-ldp`	Configures a per-prefix LFA FRR path that redirects traffic to a remote LFA tunnel via MPLS-LDP. • Use the **area** *area-id* keyword and argument to specify an area in which to enable LFA FRR.

Configuring the Maximum Distance to a Tunnel Endpoint

Perform this task to configure the maximum distance to the tunnel endpoint in a per-prefix LFA FRR path that redirects traffic to a remote LFA tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **fast-reroute per-prefix remote-lfa** [**area** *area-id*] **maximum-cost** *distance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 10`	Enables OSPF routing and enters router configuration mode.
Step 4	**fast-reroute per-prefix remote-lfa** [**area** *area-id*] **maximum-cost** *distance* **Example:** `Device(config-router)# fast-reroute per-prefix remote-lfa area 2 maximum-cost 30`	Configures the maximum distance to the tunnel endpoint in a per-prefix LFA FRR path that redirects traffic to a remote LFA tunnel. • Use the **area** *area-id* keyword and variable to specify an area in which to enable LFA FRR.

Verifying Tunnel Interfaces Created by OSPF IPv4 Remote LFA IPFRR

SUMMARY STEPS

1. **enable**
2. **show ip ospf fast-reroute remote-lfa tunnels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip ospf fast-reroute remote-lfa tunnels** **Example:** `Device# show ip ospf fast-reroute remote-lfa tunnels`	Displays information about the OSPF per-prefix LFA FRR configuration.

Configuration Examples for OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

Example: Configuring a Remote LFA Tunnel

The following example shows how to configure a remote per-prefix LFA FRR in area 2. The remote tunnel type is specified as MPLS-LDP:

```
Router(config-router)# fast-reroute per-prefix remote-lfa area 2 tunnel mpls-ldp
```

Example: Configuring the Maximum Distance to a Tunnel Endpoint

The following example shows how to set a maximum cost of 30 in area 2:

```
Router(config-router)# fast-reroute per-prefix remote-lfa area 2 maximum-cost 30
```

Example: Verifying Tunnel Interfaces Created by OSPF IPv4 Remote LFA IPFRR

The following example displays information about about tunnel interfaces created by OSPF IPv4 LFA IPFRR:

```
Router# show ip ospf fast-reroute remote-lfa tunnels

      OSPF Router with ID (192.168.1.1) (Process ID 1)
      Area with ID (0)
      Base Topology (MTID 0)

Interface MPLS-Remote-Lfa3
 Tunnel type: MPLS-LDP
 Tailend router ID: 192.168.3.3
 Termination IP address: 192.168.3.3
 Outgoing interface: Ethernet0/0
 First hop gateway: 192.168.14.4
 Tunnel metric: 20
 Protects:
  192.168.12.2 Ethernet0/1, total metric 30
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
OSPF commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: OSPF Command Reference*
Configuring OSPF	"Configuring OSPF" in the *IP Routing: OSPF Configuration Guide*.
OSPFv2 loop-free alternate fast reroute	"OSPFv2 Loop-Free Alternate Fast Reroute" in the *IP Routing: OSPF Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 5286	*Basic Specification for IP Fast Reroute: Loop-Free Alternates*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 296: Feature Information for OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

Feature Name	Releases	Feature Information
OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute	15.2(2)S Cisco IOS XE Release 3.11S	The OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute feature enables a backup repair path in the event of node failure, even if the path is multiple hops away. The following commands were introduced or modified: **fast-reroute per-prefix remote-lfa maximum-cost**, **fast-reroute per-prefix remote-lfa tunnel**, and **show ip ospf fast-reroute**.

Table 297: Feature Information for OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute

Feature Name	Releases	Feature Information
OSPF IPv4 Remote Loop-Free Alternate IP Fast Reroute	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 244

Prerequisites for OSPFv3 Multiarea Adjacency

- Ensure that Open Shortest Path First version 3 (OSPFv3) is configured on the primary interface.
- Ensure that the primary interface type is point-to-point.

Restrictions for OSPFv3 Multiarea Adjacency

- A multiarea interface operates only if OSPFv3 is configured on the primary interface and the OSPFv3 network type of the primary interface is point-to-point.
- A multiarea interface exists as a logical construct over a primary interface for OSPFv3; however, the neighbor state on the primary interface is independent of the multiarea interface.
- A multiarea interface establishes a neighbor relationship with the corresponding multiarea interface on the neighboring device. A mixture of multiarea and primary interfaces is not supported.
- A multiarea interface advertises a point-to-point connection to another device in the device link-state advertisement (LSA) for the corresponding area when the neighbor state is full.
- A multiarea interface inherits all the OSPFv3 parameters (such as, authentication) from the primary interface. You cannot configure the parameters on a multiarea interface; however, you can configure the parameters on the primary interface.

Information About OSPFv3 Multiarea Adjacency

OSPFv3 Multiarea Adjacency Overview

Open Shortest Path First version 3 (OSPFv3) allows a single physical link to be shared by multiple areas. This creates an intra-area path in each of the corresponding areas sharing the same link. All areas have an interface on which you can configure OSPFv3. One of these interfaces is designated as the primary interface and others as secondary interfaces.

The OSPFv3 Multiarea Adjacency feature allows you to configure a link on the primary interface to enable optimized routing in multiple areas. Each multiarea interface is announced as a point-to-point unnumbered link. The multiarea interface exists as a logical construct over an existing primary interface. The neighbor state on the primary interface is independent of the neighbor state of the multiarea interface. The multiarea interface establishes a neighbor relationship with the corresponding multiarea interface on the neighboring device. You can only configure multiarea adjacency on an interface that has two OSPFv3 speakers.

Use the **ospfv3 multi-area** command to configure multiarea adjacency on the primary OSPFv3 interface.

How to Configure OSPFv3 Multiarea Adjacency

Configuring OSPFv3 Multiarea Adjacency

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 enable**
5. **ospfv3 multi-area** *multi-area-id*
6. **ospfv3 multi-area** *multi-area-id* **cost** *interface-cost*
7. **ospfv3** *process-id* **ipv6 area** *area-id*
8. **serial restart-delay** *count*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial 2/0`	Specifies the interface type and number.
Step 4	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

	Command or Action	Purpose
Step 5	**ospfv3 multi-area** *multi-area-id* **Example:** `Device(config-if)# ospfv3 multi-area 100`	Configures multiarea adjacency on the interface. • The *multi-area-id* argument identifies the OSPFv3 multiarea. The range is from 0 to 4294967295, or you can use an IP address.
Step 6	**ospfv3 multi-area** *multi-area-id* **cost** *interface-cost* **Example:** `Device(config-if)# ospfv3 multi-area 100 cost 512`	(Optional) Specifies the cost of sending a packet on an OSPFv3 multiarea interface. Use this command to specify the cost only if you want the cost of the multiarea interface to be different than the cost of the primary interface.
Step 7	**ospfv3** *process-id* **ipv6 area** *area-id* **Example:** `Device(config-if)# ospfv3 1 ipv6 area 0`	Configures the OSPFv3 interface. • The *process-id* argument identifies the OSPF process. The range is from 1 to 65535. • The *area-id* argument identifies the OSPF area. The range is from 0 to 4294967295, or you can use an IP address.
Step 8	**serial restart-delay** *count* **Example:** `Device(config-if)# serial restart-delay 0`	Sets the amount of time that the router waits before trying to bring up a serial interface when it goes down. The *count* argument specifies the frequency (in seconds) at which that hardware is reset. The range is from 0 to 900.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying OSPFv3 Multiarea Adjacency

SUMMARY STEPS

1. **enable**
2. **show ospfv3 interface brief**
3. **show ospfv3 multi-area**
4. **show ospfv3 interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ospfv3 interface brief** **Example:**	Displays brief information about Open Shortest Path First version 3 (OSPFv3) interfaces.

	Command or Action	Purpose
	`Device# show ospfv3 interface brief`	
Step 3	**show ospfv3 multi-area** **Example:** `Device# show ospfv3 multi-area`	Displays information about OSPFv3 multiarea interfaces.
Step 4	**show ospfv3 interface** **Example:** `Device# show ospfv3 interface`	Displays information about OSPFv3 interfaces.

Configuration Examples for OSPFv3 Multiarea Adjacency

Example: OSPFv3 Multiarea Adjacency Configuration

```
Device> enable
Device# configure terminal
Device(config)# interface serial 2/0
Device(config-if)# ipv6 enable
Device(config-if)# ospfv3 multi-area 100
Device(config-if)# ospfv3 multi-area 100 cost 512
Device(config-if)# ospfv3 1 ipv6 area 0
Device(config-if)# serial restart-delay 0
Device(config-if)# end
```

Example: Verifying OSPFv3 Multiarea Adjacency

Sample Output for the show ospfv3 interface brief Command

To display brief information about Open Shortest Path First version 3 (OSPFv3) interfaces, use the **show ospfv3 interface brief** command in privileged EXEC mode.

```
Device# show ospfv3 interface brief

Interface PID Area  AF   Cost  State Nbrs F/C
Se2/0     1   0    ipv6  64    P2P   1/1
MA2 1     1   100  ipv6  512   P2P   1/1
```

Sample Output for the show ospfv3 multi-area Command

To display information about OSPFv3 multiarea interfaces, use the **show ospfv3 multi-area** command in privileged EXEC mode.

```
Device# show ospfv3 multi-area

OSPFV3_MA2 is up, line protocol is up
Primary Interface Serial2/0, Area 100
Interface ID 10
```

```
MTU is 1500 bytes
Neighbor Count is 1
```

Sample Output for the show ospfv3 interface Command

To display information about OSPFv3 interfaces, use the **show ospfv3 interface** command in privileged EXEC mode.

```
Device# show ospfv3 interface

Serial2/0 is up, line protocol is up
Link Local Address 2001:DB8:0:ABCD::1, Interface ID 10
Area 0, Process ID 1, Instance ID 0, Router ID 10.0.0.12
Network Type POINT_TO_POINT, Cost: 64
Transmit Delay is 1 sec, State POINT_TO_POINT
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:07
Graceful restart helper support enabled
Index 1/1/1, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 10.0.0.22
Suppress hello for 0 neighbor(s)
Multi-area interface Count is 1
OSPFV3_MA2 interface exists in area 100 Neighbor Count is 1
OSPFV3_MA2 is up, line protocol is up
Link Local Address 2001:DB8:0:ABCD::1, Interface ID 10
Area 100, Process ID 1, Instance ID 0, Router ID 10.0.0.12
Network Type POINT_TO_POINT, Cost: 512
Transmit Delay is 1 sec, State POINT_TO_POINT
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:08
Graceful restart helper support enabled
Index 1/1/2, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 10.0.0.22
```

Additional References for OSPFv3 Multiarea Adjacency

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for OSPFv3 Multiarea Adjacency

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
OSPFv3 Multiarea Adjacency	Cisco IOS XE Release 3.11S	The OSPFv3 Multiarea Adjacency feature allows you to configure a link that multiple Open Shortest Path First version 3 (OSPFv3) areas can share to enable optimized routing. You can add more than one area to an existing OSPFv3 primary interface.

Feature Name	Releases	Feature Information
OSPFv3 Multiarea Adjacency	Cisco IOS XE Release 17.4	This feature was introduced.

CHAPTER 245

OSPF Limiting Adjacency Formations

The OSPF: Limit Simultaneous Adjacency Formations feature allows you to limit to the number of adjacencies in an OSPF area.

Information About OSPF Limiting Adjacency Formations

Overview of Limiting Adjacencies

The OSPF: Limit Simultaneous Adjacency Formations feature allows you to limit to the number of adjacencies that are in “exchange” or “loading” state at the same time. A process limit (PL) determines the number of “forming” adjacencies and applies to all adjacencies for the entire process. The term “forming” refers to adjacencies that are in “exchange” or “loading” state. Adjacencies form in an OSPF area during the initial period after the area is created. The Initial Limit applies when no adjacencies have reached the “full” state in an OSPF area. If there are any “full” adjacencies in the area, the new adjacencies are governed by the Process Limit. At a given point of time, process limit and initial limit are effective in an OSPF area. When there are no adjacencies “forming” in an area, at least one adjacency is allowed to form regardless of the maximum limit specified for it. In other words, the maximum number of adjacencies can be exceeded before adjacencies form in one or more areas. The maximum limit can be exceeded by the number of areas minus one.

When a limit is reached, adjacencies in a state less than EXCHANGE are terminated. To terminate the adjacency, a hello packet is sent to the neighbor which does not have the neighbor’s device ID. This causes the neighbor to put the adjacency in the INIT state. This prevents a deadlock with the neighbor, which could otherwise happen if the neighbor is blocking an adjacency from forming on a different interface. By causing the neighbor to bring the adjacency to INIT, it allows the neighbor to form an adjacency on a different interface. Packets from unknown neighbors are ignored when the limit has been reached or exceeded.

If graceful restart or Cisco nonstop forwarding is configured, the hello packets must be accepted from every neighboring device The restarting device must include the neighbors’ device IDs in its hello packets to prevent the adjacency from being dropped by the neighbor. If graceful restart is in configured, the grace link-state advertisements (LSAs) must be sent in a normal mode and not in a throttling mode. When the device is performing graceful restart and if the limit is reached, new adjacencies are allowed to remain in 2-WAY or

EXSTART. However, they are prevented from proceeding to EXCHANGE until the number of forming adjacencies is less than the limit.

Configuring Adjacency Formations

Use the **adjacency stagger** command to configure the maximum limit and the initial limit for an area in the router or address-family configuration modes. The initial limit must not be greater than the process limit. The default value is 300 and the minimum is 1. If the **none** keyword is used, the maximum limit is only effective. The **none** keyword also disables the initial limit for areas. If an initial limit is reached in an area and no adjacencies are forming, no adjacencies will be allowed to form in the area until global number of adjacencies forming is less than the PL.

Use the **ip ospf adjacency stagger disable** or the **ospfv3 adjacency stagger disable** command to disable staggering on an interface. Adjacencies forming on a disabled interface are counted towards throttling limits. Disabling the throttling on an interface allows exceeding the maximum limit when the maximum limit is reached and a new adjacency forms on an interface where throttling is disabled.

Note When using the **no adjacency stagger** command to disable the feature, the command is displayed in the running configuration. To return to the default values, use the **default adjacency stagger** command. After using this command, the **adjacency stagger** command does not appear in the running configuration.

How to Configure OSPF Limiting Adjacency Formations

Configuring Adjacency Formations Globally

Configuring Adjacency Limit in the Router Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id*
4. **adjacency stagger** {*initial-limit* | **none**} *maximum-limit*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router ospf** *process-id* **Example:** `Device(config)# router ospf 109`	Enables OSPF routing and enters router configuration mode.
Step 4	**adjacency stagger** {*initial-limit* | **none**} *maximum-limit* **Example:** `Device(config-router)# adjacency stagger 10 50`	Controls the number of adjacencies forming in an area. • *initial-limit*—Minimum number of adjacencies allowed in an area. • *maximum-limit*—Maximum number of adjacencies allowed in an area. • **none**—No minimum number for adjacencies allowed in an area.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Adjacency Limit in the Address Family Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospfv3** [*process-id*]
4. Do one of the following:
   - **address-family ipv4 unicast**
   - **address-family ipv6 unicast**
5. **adjacency stagger** {*initial-limit* | **none**} {*maximum-limit*| **disable**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**router ospfv3** [*process-id*] **Example:** `Device(config)# router ospfv3 1`	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.

Step 4	Do one of the following:
	• **address-family ipv4 unicast**
	• **address-family ipv6 unicast**
	Example:
	`Device(config-router)# address-family ipv4 unicast`
	Example:
	`Device(config-router)# address-family ipv6 unicast`
		Enters IPv4 or IPv6 address family configuration mode for OSPFv3.

Step 5	**adjacency stagger** {*initial-limit* | **none**} {*maximum-limit*| **disable**} **Example:** `Device(config-router-af)# adjacency stagger 10 50`	Controls the number of adjacencies forming in an area. • *initial-limit*—Minimum number of adjacencies allowed in an area. • **none**—No minimum number for adjacencies allowed in an area. • *maximum-limit*—Maximum number of adjacencies allowed in an area. • **disable**—Disable adjacency formations.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Disabling Adjacency Staggering in the Interface Configuration Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ip ospf adjacency stagger disable**
   - **ospfv3 adjacency stagger disable**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface serial 2/0`	Specifies the interface and enters interface configuration mode.

Step 4	Do one of the following:
	• **ip ospf adjacency stagger disable**
	• **ospfv3 adjacency stagger disable**
	Example:
	`Device(config-if)# ip ospf adjacency stagger disable`
	Example:
	`Device(config-if)# ospfv3 adjacency stagger disable`
		Disables adjacency staggering on the interface.

Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Verifying Adjacency Staggering

SUMMARY STEPS

1. **enable**
2. **show ip ospf**
3. **show ospfv3**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show ip ospf**

Example:

```
Device# show ip ospf

 Routing Process "ospf 10" with ID 10.8.3.3
 Start time: 2w0d, Time elapsed: 00:16:43.033
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Supports NSSA (compatible with RFC 3101)
 Supports Database Exchange Summary List Optimization (RFC 5243)
 Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 EXCHANGE/LOADING adjacency limit: initial 300, process maximum 300
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
 IETF NSF helper support enabled
 Cisco NSF helper support enabled
 Reference bandwidth unit is 100 mbps
```

Displays information about OSPF routing processes.

Step 3 **show ospfv3**

Example:

```
Device# show ospfv3

 OSPFv3 12 address-family ipv6
 Router ID 10.8.3.3
 Supports NSSA (compatible with RFC 3101)
 Supports Database Exchange Summary List Optimization (RFC 5243)
 Event-log enabled, Maximum number of events: 1000, Mode: cyclic
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Retransmission limit dc 24 non-dc 24
 EXCHANGE/LOADING adjacency limit: initial 10, process maximum 50
 Number of external LSA 0. Checksum Sum 0x000000
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 Graceful restart helper support enabled
```

```
Reference bandwidth unit is 100 mbps
RFC1583 compatibility enabled
```

Displays information about OSPFv3 routing processes.

Configuration Examples for OSPF Limiting Adjacency Formations

Example: Configuring Adjacency Limit in the Router Configuration Mode

```
Device> enable
Device# configure terminal
Device(config)# router ospf 109
Device(config-router)# adjacency stagger 10 50
Device(config-router)# end
```

Example: Configuring Adjacency Limit in the Address Family Configuration Mode

```
Device> enable
Device# configure terminal
Device(config)# router ospfv3 1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# adjacency stagger 10 50
Device(config-router-af)# end
```

Example: Disabling Adjacency in the Interface Configuration Mode

```
Device> enable
Device# configure terminal
Device(config)# interface serial 2/0
Device(config-if)# ospfv3 adjacency stagger disable
Device(config-if)# end
```

Additional References for OSPF Limiting Adjacency Formations

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference
Configuring OSPF	Configuring OSPF

Related Topic	Document Title
Multiarea Adjacency	• OSPFv2 Multiarea Adjacency • OSPFv3 Multiarea Adjacency

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPF Limiting Adjacencies Formations

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 298: Feature Information for OSPF Limiting Adjacencies Formations

Feature Name	Releases	Feature Information
OSPF: Limit Simultaneous Adjacency Formations	Cisco IOS XE Release 3.15S	The following commands were introduced or modified: **adjacency stagger**, **ip ospf adjacency stagger disable**, **ip ospfv3 adjacency stagger disable**, **show ip ospf**, **show ip ospfv3**.

Table 299: Feature Information for OSPF Limiting Adjacencies Formations

Feature Name	Releases	Feature Information
OSPF: Limit Simultaneous Adjacency Formations	Cisco IOS XE Release 17.4	This feature was introduced.

PART VIII

RIP

CHAPTER 246

IPv6 Routing: RIP for IPv6

IPv6 Routing Information Protocol (RIP) functions the same and offers the same benefits as IPv4 RIP. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes and the use of the all-RIP-devices multicast group address, FF02::9, as the destination address for RIP update messages.

Information About RIP for IPv6

RIP for IPv6

IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages.

In the Cisco software implementation of IPv6 RIP, each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB). The IPv6 RIP RIB contains a set of best-cost IPv6 RIP routes learned from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it will store only the lowest cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors running RIP. IPv6 RIP will try to insert every non-expired route from its local RIB into the primary IPv6 RIB. If the same route has been learned from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route will not be added to the IPv6 RIB but the RIP route will still exist in the IPv6 RIP RIB.

Nonstop Forwarding for IPv6 RIP

Cisco nonstop forwarding (NSF) continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. When an RP failover occurs, the Forwarding Information Base (FIB) marks installed paths as stale by setting a new epoch. Subsequently, the routing protocols reconverge and populate the RIB and FIB. Once all NSF routing protocols converge, any stale routes held in the FIB are removed. A failsafe timer is required to delete stale routes, in case of routing protocol failure to repopulate the RIB and FIB.

RIP registers as an IPv6 NSF client. Doing so has the benefit of using RIP routes installed in the Cisco Express Forwarding table until RIP has converged on the standby.

How to Configure RIP for IPv6

Enabling IPv6 RIP

Before you begin

Before configuring the router to run IPv6 RIP, globally enable IPv6 using the **ipv6 unicast-routing** command in global configuration mode, and enable IPv6 on any interfaces on which IPv6 RIP is to be enabled.

If you want to set or change a global value, follow steps 1 and 2, and then use the optional **ipv6 router rip**command in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 rip** *name* **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**ipv6 rip** *name* **enable** **Example:** `Router(config-if)# ipv6 rip process1 enable`	Enables the specified IPv6 RIP routing process on an interface.

Customizing IPv6 RIP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router rip** *word*
4. **maximum-paths** *number-paths*
5. **exit**
6. **interface** *type number*
7. **ipv6 rip** *name* **default-information** {**only** | **originate**} [**metric** *metric-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router rip** *word* **Example:** `Router(config)# ipv6 router rip process1`	Configures an IPv6 RIP routing process and enters router configuration mode for the IPv6 RIP routing process. • Use the *word*argument to identify a specific IPv6 RIP routing process.
Step 4	**maximum-paths** *number-paths* **Example:** `Router(config-router)# maximum-paths 1`	(Optional) Defines the maximum number of equal-cost routes that IPv6 RIP can support. • The *number-paths* argument is an integer from 1 to 64. The default for RIP is four paths.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 7	**ipv6 rip** *name* **default-information** {**only** | **originate**} [**metric** *metric-value*] **Example:** `Router(config-if)# ipv6 rip process1 default-information originate`	(Optional) Originates the IPv6 default route (::/0) into the specified RIP routing process updates sent out of the specified interface. **Note** To avoid routing loops after the IPv6 default route (::/0) is originated out of any interface, the routing process ignores all default routes received on any interface. • Specifying the **only** keyword originates the default route (::/0) but suppresses all other routes in the updates sent on this interface. • Specifying the **originate** keyword originates the default route (::/0) in addition to all other routes in the updates sent on this interface.

Verifying IPv6 RIP Configuration and Operation

SUMMARY STEPS

1. **show ipv6 rip** [*name*][**database**| **next-hops**]
2. **show ipv6 route** [*ipv6-address*| *ipv6-prefix*/*prefix-length*| *protocol* | *interface-type interface-number*]
3. **enable**
4. **debug ipv6 rip** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ipv6 rip** [*name*][**database**| **next-hops**] **Example:** `Device> show ipv6 rip process1 database`	(Optional) Displays information about current IPv6 RIP processes. • In this example, IPv6 RIP process database information is displayed for the specified IPv6 RIP process.
Step 2	**show ipv6 route** [*ipv6-address*| *ipv6-prefix*/*prefix-length*| *protocol*	*interface-type interface-number*] **Example:** `Device> show ipv6 route rip`	(Optional) Displays the current contents of the IPv6 routing table. • In this example, only IPv6 RIP routes are displayed.

	Command or Action	Purpose
Step 3	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 4	**debug ipv6 rip** [*interface-type interface-number*] **Example:** `Device# debug ipv6 rip`	(Optional) Displays debugging messages for IPv6 RIP routing transactions.

Configuration Examples for RIP for IPv6

Example: Enabling the RIP for IPv6 Process

In the following example, the IPv6 RIP process named process1 is enabled on the router and on Gigabit Ethernet interface 0/0/0. The IPv6 default route (::/0) is advertised in addition to all other routes in router updates sent on Gigabit Ethernet interface 0/0/0. Additionally, BGP routes are redistributed into the RIP process named process1 according to a route map where routes that match a prefix list are also tagged. The number of parallel paths is set to one to allow the route tagging, and the IPv6 RIP timers are adjusted. A prefix list named eth0/0-in-flt filters inbound routing updates on Gigabit Ethernet interface 0/0/0.

```
ipv6 router rip process1
 maximum-paths 1
 redistribute bgp 65001 route-map bgp-to-rip
 distribute-list prefix-list eth0/0-in-flt in Gigabitethernet0/0/0
!
interface Gigabitethernet0/0/0
 ipv6 address 2001:DB8::/64 eui-64
 ipv6 rip process1 enable
 ipv6 rip process1 default-information originate
!
ipv6 prefix-list bgp-to-rip-flt seq 10 deny 2001:DB8:3::/16 le 128
ipv6 prefix-list bgp-to-rip-flt seq 20 permit 2001:DB8:1::/8 le 128
!
ipv6 prefix-list eth0/0-in-flt seq 10 deny ::/0
ipv6 prefix-list eth0/0-in-flt seq 15 permit ::/0 le 128
!
route-map bgp-to-rip permit 10
 match ipv6 address prefix-list bgp-to-rip-flt
 set tag 4
```

In the following example, output information about all current IPv6 RIP processes is displayed using the **show ipv6 rip** command:

```
Device> show ipv6 rip

RIP process "process1", port 521, multicast-group FF02::9, pid 62
     Administrative distance is 120. Maximum paths is 1
     Updates every 5 seconds, expire after 15
     Holddown lasts 10 seconds, garbage collect after 30
     Split horizon is on; poison reverse is off
```

```
    Default routes are generated
    Periodic updates 223, trigger updates 1
  Interfaces:
    Gigabitethernet0/0/0
  Redistribution:
    Redistributing protocol bgp 65001 route-map bgp-to-rip
```

In the following example, output information about a specified IPv6 RIP process database is displayed using the **show ipv6 rip** command with the *name* argument and the **database** keyword. In the following output for the IPv6 RIP process named process1, timer information is displayed, and route 2001:DB8::16/64 has a route tag set:

```
Device> show ipv6 rip process1 database

RIP process "process1", local RIB
 2001:DB8::/64, metric 2
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:1::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:2::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 ::/0, metric 2, installed
     Gigabitethernet0/0/0FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
```

In the following example, output information for a specified IPv6 RIP process is displayed using the **show ipv6 rip** command with the *name* argument and the **next-hops** keyword:

```
Device> show ipv6 rip process1 next-hops

RIP process "process1", Next Hops
  FE80::A8BB:CCFF:FE00:A00/Gigabitethernet0/0/0 [4 paths]
```

Additional References

The following sections provide references related to configuring Routing Information Protocol.

Related Documents

Related Topic	Document Title
Protocol-independent features, filtering RIP information, key management (available in RIP Version 2), and VLSM	*Configuring IP Routing Protocol-Independent Features*
IPv6 Routing: RIP for IPv6	*Cisco IOS IP Routing: RIP Configuration Guide*
RIP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: RIP Command Reference*
Configuring Frame Relay	*Cisco IOS Wide-Area Networking Configuration Guide*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBS are supported and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1058	*Routing Information Protocol*
RFC 2082	RIP-2 MD5 Authentication
RFC 2091	*Triggered Extensions to RIP to Support Demand Circuits*
RFC 2453	RIP version 2

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for RIP for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 300: Feature Information for RIP for IPv6

Feature Name	Releases	Feature Information
IPv6 Routing: RIP for IPv6 (RIPng)	12.2(2)T 12.2(17a)SX1 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 12.3 15.0(2)SG Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2.0SG	RIP enhancements for IPv6 include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages. The following commands were introduced or modified: **debug ipv6 rip**, **ipv6 rip default-information**, **ipv6 rip enable**, **ipv6 router rip**, **ipv6 unicast-routing**, **maximum-paths**, **show ipv6 rip**, **show ipv6 route**.
IPv6: RIPng Nonstop Forwarding	12.2(33)SRE 15.0(1)S 15.0(1)SY Cisco IOS XE Release 2.1	The IPv6 RIPng nonstop forwarding feature is supported.

CHAPTER 247

IPv6 Routing: Route Redistribution

IPv6 route redistribution allows routes to be specified by prefix, using a route-map prefix list, or by tag, using the route-map "match tag" function.

Information About IPv6 Route Redistribution

RIP for IPv6

IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages.

In the Cisco software implementation of IPv6 RIP, each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB). The IPv6 RIP RIB contains a set of best-cost IPv6 RIP routes learned from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it will store only the lowest cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors running RIP. IPv6 RIP will try to insert every non-expired route from its local RIB into the primary IPv6 RIB. If the same route has been learned from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route will not be added to the IPv6 RIB but the RIP route will still exist in the IPv6 RIP RIB.

How to Configure IPv6 Route Redistribution

Redistributing Routes into an IPv6 RIP Routing Process

The maximum metric that RIP can advertise is 16, and a metric of 16 denotes a route that is unreachable. Therefore, if you are redistributing routes with metrics greater than or equal to 16, then by default RIP will

advertise them as unreachable. These routes will not be used by neighboring routers. The user must configure a redistribution metric of less than 15 for these routes.

Note You must to advertise a route with metric of 15 or less. A RIP router always adds an interface cost--the default is 1--onto the metric of a received route. If you advertise a route with metric 15, your neighbor will add 1 to it, making a metric of 16. Because a metric of 16 is unreachable, your neighbor will not install the route in the routing table.

If no metric is specified, then the current metric of the route is used. To find the current metric of the route, enter the **show ipv6 route** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 rip** *word* **enable**
5. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2**| **level-2**} [**metric** *metric-value*] [**metric-type**{**internal** | **external**}] [**route-map** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ipv6 rip** *word* **enable** **Example:** `Router(config-if)# ipv6 router one enable`	Enables an IPv6 Routing Information Protocol (RIP) routing process on an interface.
Step 5	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2**| **level-2**} [**metric** *metric-value*] [**metric-type**{**internal** | **external**}] [**route-map** *map-name*] **Example:**	Redistributes the specified routes into the IPv6 RIP routing process. • The *protocol* argument can be one of the following keywords: **bgp**, **connected**, **isis**, **rip**, or **static**.

	Command or Action	Purpose
	`Router(config-router)# redistribute bgp 65001 route-map bgp-to-rip`	• The **rip** keyword and *process-id* argument specify an IPv6 RIP routing process. **Note** The **connected** keyword refers to routes that are established automatically by assigning IPv6 addresses to an interface.

Configuring Route Tags for IPv6 RIP Routes

When performing route redistribution, you can associate a numeric tag with a route. The tag is advertised with the route by RIP and will be installed along with the route in neighboring router's routing table.

If you redistribute a tagged route (for example, a route in the IPv6 routing table that already has a tag) into RIP, then RIP will automatically advertise the tag with the route. If you use a redistribution route map to specify a tag, then RIP will use the route map tag in preference to the routing table tag.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*
5. **set tag** *tag-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-tag* [**permit**	**deny**] [*sequence-number*] **Example:** `Router(config)# route-map bgp-to-rip permit 10`	Defines a route map, and enters route-map configuration mode. • Follow this step with a **match** command.
Step 4	**match ipv6 address** {**prefix-list** *prefix-list-name*	*access-list-name* **Example:**	Specifies a list of IPv6 prefixes to be matched.

	Command or Action	Purpose
	`Router(config-route-map)# match ipv6 address prefix-list bgp-to-rip-flt`	
Step 5	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 4`	Sets the tag value to associate with the redistributed routes.

Filtering IPv6 RIP Routing Updates

Route filtering using distribute lists provides control over the routes RIP receives and advertises. This control may be exercised globally or per interface.

Filtering is controlled by distribute lists. Input distribute lists control route reception, and input filtering is applied to advertisements received from neighbors. Only those routes that pass input filtering will be inserted in the RIP local routing table and become candidates for insertion into the IPv6 routing table.

Output distribute lists control route advertisement; Output filtering is applied to route advertisements sent to neighbors. Only those routes passing output filtering will be advertised.

Global distribute lists (which are distribute lists that do not apply to a specified interface) apply to all interfaces. If a distribute list specifies an interface, then that distribute list applies only to that interface.

An interface distribute list always takes precedence. For example, for a route received at an interface, with the interface filter set to deny, and the global filter set to permit, the route is blocked, the interface filter is passed, the global filter is blocked, and the route is passed.

IPv6 prefix lists are used to specify certain prefixes or a range of prefixes that must be matched before a permit or deny statement can be applied. Two operand keywords can be used to designate a range of prefix lengths to be matched. A prefix length of less than, or equal to, a value is configured with the **le**keyword. A prefix length greater than, or equal to, a value is specified using the **ge** keyword. The **ge** and **le** keywords can be used to specify the range of the prefix length to be matched in more detail than the usual *ipv6-prefix* / *prefix-length* argument. For a candidate prefix to match against a prefix list entry three conditions can exist:

- The candidate prefix must match the specified prefix list and prefix length entry.
- The value of the optional **le** keyword specifies the range of allowed prefix lengths from the *prefix-length* argument up to, and including, the value of the **le** keyword.
- The value of the optional **ge** keyword specifies the range of allowed prefix lengths from the value of the **ge** keyword up to, and including, 128.

Note Note that the first condition must match before the other conditions take effect.

An exact match is assumed when the **ge** or **le** keywords are not specified. If only one keyword operand is specified then the condition for that keyword is applied, and the other condition is not applied. The *prefix-length* value must be less than the **ge** value. The **ge** value must be less than, or equal to, the **le** value. The **le** value must be less than or equal to 128.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*
4. **ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*
5. Repeat Steps 3 and 4 as many times as necessary to build the prefix list.
6. **ipv6 router rip** *name*
7. **distribute-list prefix-list** *prefix-list-name* **in** | **out**} [*interface-type interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length*	**description** *text*} [**ge** *ge-value*] [**le** *le-value* **Example:** `Router(config)# ipv6 prefix-list abc permit 2001:DB8::/16`	Creates an entry in the IPv6 prefix list.
Step 4	**ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length*	**description** *text*} [**ge** *ge-value*] [**le** *le-value* **Example:** `Router(config)# ipv6 prefix-list abc deny ::/0`	Creates an entry in the IPv6 prefix list.
Step 5	Repeat Steps 3 and 4 as many times as necessary to build the prefix list.	--
Step 6	**ipv6 router rip** *name* **Example:** `Router(config)# ipv6 router rip process1`	Configures an IPv6 RIP routing process.

	Command or Action	Purpose
Step 7	**distribute-list prefix-list** *prefix-list-name* **in** | **out**} [*interface-type interface-number* **Example:** `Router(config-rtr-rip)# distribute-list prefix-list process1 in gigabitethernet 0/0/0`	Applies a prefix list to IPv6 RIP routing updates that are received or sent on an interface.

Configuration Examples for IPv6 Route Redistribution

Example: Enabling the RIP for IPv6 Process

In the following example, the IPv6 RIP process named process1 is enabled on the router and on Gigabit Ethernet interface 0/0/0. The IPv6 default route (::/0) is advertised in addition to all other routes in router updates sent on Gigabit Ethernet interface 0/0/0. Additionally, BGP routes are redistributed into the RIP process named process1 according to a route map where routes that match a prefix list are also tagged. The number of parallel paths is set to one to allow the route tagging, and the IPv6 RIP timers are adjusted. A prefix list named eth0/0-in-flt filters inbound routing updates on Gigabit Ethernet interface 0/0/0.

```
ipv6 router rip process1
 maximum-paths 1
 redistribute bgp 65001 route-map bgp-to-rip
 distribute-list prefix-list eth0/0-in-flt in Gigabitethernet0/0/0
!
interface Gigabitethernet0/0/0
 ipv6 address 2001:DB8::/64 eui-64
 ipv6 rip process1 enable
 ipv6 rip process1 default-information originate
!
ipv6 prefix-list bgp-to-rip-flt seq 10 deny 2001:DB8:3::/16 le 128
ipv6 prefix-list bgp-to-rip-flt seq 20 permit 2001:DB8:1::/8 le 128
!
ipv6 prefix-list eth0/0-in-flt seq 10 deny ::/0
ipv6 prefix-list eth0/0-in-flt seq 15 permit ::/0 le 128
!
route-map bgp-to-rip permit 10
 match ipv6 address prefix-list bgp-to-rip-flt
 set tag 4
```

In the following example, output information about all current IPv6 RIP processes is displayed using the **show ipv6 rip** command:

```
Device> show ipv6 rip

RIP process "process1", port 521, multicast-group FF02::9, pid 62
     Administrative distance is 120. Maximum paths is 1
     Updates every 5 seconds, expire after 15
     Holddown lasts 10 seconds, garbage collect after 30
     Split horizon is on; poison reverse is off
     Default routes are generated
     Periodic updates 223, trigger updates 1
  Interfaces:
    Gigabitethernet0/0/0
```

```
  Redistribution:
    Redistributing protocol bgp 65001 route-map bgp-to-rip
```

In the following example, output information about a specified IPv6 RIP process database is displayed using the **show ipv6 rip** command with the *name* argument and the **database** keyword. In the following output for the IPv6 RIP process named process1, timer information is displayed, and route 2001:DB8::16/64 has a route tag set:

```
Device> show ipv6 rip process1 database

RIP process "process1", local RIB
 2001:DB8::/64, metric 2
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:1::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:2::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 ::/0, metric 2, installed
     Gigabitethernet0/0/0FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
```

In the following example, output information for a specified IPv6 RIP process is displayed using the **show ipv6 rip** command with the *name* argument and the **next-hops** keyword:

```
Device> show ipv6 rip process1 next-hops

RIP process "process1", Next Hops
  FE80::A8BB:CCFF:FE00:A00/Gigabitethernet0/0/0 [4 paths]
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: Route Redistribution

Table 301: Feature Information for IPv6 Routing: Route Redistribution

Feature Name	Releases	Feature Information
IPv6 Routing: Route Redistribution	Cisco IOS XE Release 2.1	Routes may be specified by prefix, using a route-map prefix list, or by tag, using the route-map "match tag" function. The following commands were introduced or modified: **distribute-list prefix-list**, **ipv6 prefix list**, **ipv6 rip enable**, **ipv6 router rip**, **match ipv6 address**, **redistribute**, **route-map**, **set tag**, **show ipv6 rip**.

CHAPTER 248

Configuring Routing Information Protocol

Routing Information Protocol (RIP) is a commonly used routing protocol in small to medium TCP/IP networks. It is a stable protocol that uses a distance-vector algorithm to calculate routes.

Prerequisites for RIP

You must configure **ip routing** command before you configure RIP.

Restrictions for RIP

Routing Information Protocol (RIP) uses hop count as the metric to rate the value of different routes. The hop count is the number of devices that can be traversed in a route. A directly connected network has a metric of zero; an unreachable network has a metric of 16. This limited metric range makes RIP unsuitable for large networks.

Note If RIP configuration does not have a network statement covering a specific interface, we recommend that you do not configure RIP under that interface. If RIP is configured on such an interface, the redistribution of route(s) from another routing protocol into RIP, received through that interface, does not work.

Information About Configuring RIP

RIP Overview

The Routing Information Protocol (RIP) uses broadcast UDP data packets to exchange routing information. Cisco software sends routing information updates every 30 seconds, which is termed advertising. If a device does not receive an update from another device for 180 seconds or more, the receiving device marks the routes served by the nonupdating device as unusable. If there is still no update after 240 seconds, the device removes all routing table entries for the nonupdating device.

A device that is running RIP can receive a default network via an update from another device that is running RIP, or the device can source the default network using RIP. In both cases, the default network is advertised through RIP to other RIP neighbors.

The Cisco implementation of RIP Version 2 (RIPv2) supports plain text and message digest algorithm 5 (MD5) authentication, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs).

RIP Routing Updates

The Routing Information Protocol (RIP) sends routing-update messages at regular intervals and when the network topology changes. When a device receives a RIP routing update that includes changes to an entry, the device updates its routing table to reflect the new route. The metric value for the path is increased by 1, and the sender is indicated as the next hop. RIP devices maintain only the best route (the route with the lowest metric value) to a destination. After updating its routing table, the device immediately begins transmitting RIP routing updates to inform other network devices of the change. These updates are sent independently of the regularly scheduled updates that RIP devices send.

RIP Routing Metric

The Routing Information Protocol (RIP) uses a single routing metric to measure the distance between the source and the destination network. Each hop in a path from the source to the destination is assigned a hop-count value, which is typically 1. When a device receives a routing update that contains a new or changed destination network entry, the device adds 1 to the metric value indicated in the update and enters the network in the routing table. The IP address of the sender is used as the next hop. If an interface network is not specified in the routing table, it will not be advertised in any RIP update.

Authentication in RIP

The Cisco implementation of the Routing Information Protocol (RIP) Version 2 (RIPv2) supports authentication, key management, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs).

By default, the software receives RIP Version 1 (RIPv1) and RIPv2 packets, but sends only RIPv1 packets. You can configure the software to receive and send only RIPv1 packets. Alternatively, you can configure the software to receive and send only RIPv2 packets. To override the default behavior, you can configure the RIP version that an interface sends. Similarly, you can also control how packets received from an interface are processed.

RIPv1 does not support authentication. If you are sending and receiving RIP v2 packets, you can enable RIP authentication on an interface.

The key chain determines the set of keys that can be used on the interface. Authentication, including default authentication, is performed on that interface only if a key chain is configured. For more information on key chains and their configuration, see the "Managing Authentication Keys" section in the "Configuring IP Routing Protocol-Independent Features" chapter in the *Cisco IOS IP Routing: Protocol-Independent Configuration Guide*.

Cisco supports two modes of authentication on an interface on which RIP is enabled: plain-text authentication and message digest algorithm 5 (MD5) authentication. Plain-text authentication is the default authentication in every RIPv2 packet.

Note Do not use plain text authentication in RIP packets for security purposes, because the unencrypted authentication key is sent in every RIPv2 packet. Use plain-text authentication when security is not an issue; for example, you can use plain-text authentication to ensure that misconfigured hosts do not participate in routing.

Exchange of Routing Information

Routing Information Protocol (RIP) is normally a broadcast protocol, and for RIP routing updates to reach nonbroadcast networks, you must configure the Cisco software to permit this exchange of routing information.

To control the set of interfaces with which you want to exchange routing updates, you can disable the sending of routing updates on specified interfaces by configuring the **passive-interface** router configuration command.

You can use an offset list to increase increasing incoming and outgoing metrics to routes learned via RIP. Optionally, you can limit the offset list with either an access list or an interface.

Routing protocols use several timers that determine variables such as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs. You can make the following timer adjustments:

- The rate (time, in seconds, between updates) at which routing updates are sent
- The interval of time, in seconds, after which a route is declared invalid
- The interval, in seconds, during which routing information about better paths is suppressed
- The amount of time, in seconds, that must pass before a route is removed from the routing table
- The amount of time for which routing updates will be postponed

You can adjust the IP routing support in the Cisco software to enable faster convergence of various IP routing algorithms, and hence, cause quicker fallback to redundant devices. The total effect is to minimize disruptions to end users of the network in situations where quick recovery is essential

In addition, an address family can have timers that explicitly apply to that address family (or Virtual Routing and Forwarding [VRF]) instance). The **timers-basic** command must be specified for an address family or the system defaults for the **timers-basic** command are used regardless of the timer that is configured for RIP routing. The VRF does not inherit the timer values from the base RIP configuration. The VRF will always use the system default timers unless the timers are explicitly changed using the **timers-basic** command.

RIP Route Summarization

Summarizing routes in RIP Version 2 improves scalability and efficiency in large networks. Summarizing IP addresses means that there is no entry for child routes (routes that are created for any combination of the individual IP addresses contained within a summary address) in the RIP routing table, reducing the size of the table and allowing the router to handle more routes.

Summary IP address functions more efficiently than multiple individually advertised IP routes for the following reasons:

- The summarized routes in the RIP database are processed first.
- Any associated child routes that are included in a summarized route are skipped as RIP looks through the routing database, reducing the processing time required. Cisco routers can summarize routes in two ways:
- Automatically, by summarizing subprefixes to the classful network boundary when crossing classful network boundaries (automatic summary).

Note Automatic summary is enabled by default.

- As specifically configured, advertising a summarized local IP address pool on the specified interface (on a network access server) so that the address pool can be provided to dialup clients.

When RIP determines that a summary address is required in the RIP database, a summary entry is created in the RIP routing database. As long as there are child routes for a summary address, the address remains in the routing database. When the last child route is removed, the summary entry also is removed from the database. This method of handling database entries reduces the number of entries in the database because each child route is not listed in an entry, and the aggregate entry itself is removed when there are no longer any valid child routes for it.

RIP Version 2 route summarization requires that the lowest metric of the "best route" of an aggregated entry, or the lowest metric of all current child routes, be advertised. The best metric for aggregated summarized routes is calculated at route initialization or when there are metric modifications of specific routes at advertisement time, and not at the time the aggregated routes are advertised.

The **ip summary-address rip router**configuration command causes the router to summarize a given set of routes learned via RIP Version 2 or redistributed into RIP Version 2. Host routes are especially applicable for summarization.

See the "Route Summarization Example, on page 3002" section at the end of this chapter for examples of using split horizon.

You can verify which routes are summarized for an interface using the **show ip protocols** EXEC command. You can check summary address entries in the RIP database. These entries will appear in the database only if relevant child routes are being summarized. To display summary address entries in the RIP routing database entries if there are relevant routes being summarized based upon a summary address, use the **show ip rip database** command in EXEC mode. When the last child route for a summary address becomes invalid, the summary address is also removed from the routing table.

Split Horizon Mechanism

Normally, devices that are connected to broadcast-type IP networks and that use distance-vector routing protocols employ the split horizon mechanism to reduce the possibility of routing loops. The split horizon mechanism blocks information about routes from being advertised by a device out of any interface from which that information originated. This behavior usually optimizes communications among multiple devices, particularly when links are broken. However, with nonbroadcast networks, such as Frame Relay and the Switched Multimegabit Digital System (SMDS), situations can arise for which this behavior is less than ideal. In such situations, you may want to disable split horizon with the Routing Information Protocol (RIP).

If an interface is configured with secondary IP addresses and split horizon is enabled, updates might not be sourced by the secondary address. If split horizon is enabled, one routing update is sourced per network number.

Split horizon is not disabled by default for interfaces using any of the X.25 encapsulations. For all other encapsulations, split horizon is enabled by default.

Interpacket Delay for RIP Updates

By default, the software adds no delay between packets in a multiple-packet RIP update being sent. If you have a high-end router sending to a low-speed router, you might want to add such interpacket delay to RIP updates, in the range of 8 to 50 milliseconds.

RIP Optimization over WAN Circuits

Devices are used on connection-oriented networks to allow potential connectivity to many remote destinations. Circuits on the WAN are established on demand and are relinquished when the traffic subsides. Depending on the application, the connection between any two sites for user data could be short and relatively infrequent.

Source IP Addresses of RIP Routing Updates

By default, the Cisco software validates the source IP address of incoming Routing Information Protocol (RIP) routing updates. If the source address is not valid, the software discards the routing update. You must disable this functionality if you want to receive updates from a device that is not part of this network. However, disabling this functionality is not recommended under normal circumstances.

Neighbor Router Authentication

You can prevent your router from receiving fraudulent route updates by configuring neighbor router authentication. When configured, neighbor authentication occurs whenever routing updates are exchanged between neighbor routers. This authentication ensures that a router receives reliable routing information from a trusted source.

Without neighbor authentication, unauthorized or deliberately malicious routing updates could compromise the security of your network traffic. A security compromise could occur if an unfriendly party diverts or analyzes your network traffic. For example, an unauthorized router could send a fictitious routing update to convince your router to send traffic to an incorrect destination. This diverted traffic could be analyzed to learn confidential information about your organization or merely used to disrupt your organization’s ability to effectively communicate using the network. Neighbor authentication prevents any such fraudulent route updates from being received by your router.

When neighbor authentication has been configured on a router, the router authenticates the source of each routing update packet that it receives. This is accomplished by the exchange of an authenticating key (sometimes referred to as a password) that is known to both the sending and the receiving router.

There are two types of neighbor authentication used: plain text authentication and Message Digest Algorithm Version 5 (MD5) authentication. Both forms work in the same way, with the exception that MD5 sends a "message digest" instead of the authenticating key itself. The message digest is created using the key and a message, but the key itself is not sent, preventing it from being read while it is being transmitted. Plain text authentication sends the authenticating key itself over the wire.

Note Note that plain text authentication is not recommended for use as part of your security strategy. Its primary use is to avoid accidental changes to the routing infrastructure. Using MD5 authentication, however, is a recommended security practice.

In plain text authentication, each participating neighbor router must share an authenticating key. This key is specified at each router during configuration. Multiple keys can be specified with some protocols; each key must then be identified by a key number.

In general, when a routing update is sent, the following authentication sequence occurs:

1. A router sends a routing update with a key and the corresponding key number to the neighbor router. In protocols that can have only one key, the key number is always zero. The receiving (neighbor) router checks the received key against the same key stored in its own memory.
2. If the two keys match, the receiving router accepts the routing update packet. If the two keys do not match, the routing update packet is rejected.

MD5 authentication works similarly to plain text authentication, except that the key is never sent over the wire. Instead, the router uses the MD5 algorithm to produce a "message digest" of the key (also called a "hash"). The message digest is then sent instead of the key itself. This ensures that nobody can eavesdrop on the line and learn keys during transmission.

Another form of neighbor router authentication is to configure key management using key chains. When you configure a key chain, you specify a series of keys with lifetimes, and the Cisco IOS software rotates through each of these keys. This decreases the likelihood that keys will be compromised. To find complete configuration information for key chains, refer to the "Managing Authentication Keys" section in the Configuring IP Routing Protocol-Independent Features module of the Cisco IOS IP Routing: Protocol-Independent Configuration Guide.

IP-RIP Delay Start Overview

The IP-RIP Delay Start feature is used on Cisco devices to delay the initiation of Routing Information Protocol Version 2 (RIPv2) neighbor sessions until the network connectivity between the neighbor devices is fully operational, thereby ensuring that the sequence number of the first message digest algorithm 5 (MD5) packet that the device sends to the non-Cisco neighbor device is 0. The default behavior for a device configured to establish RIPv2 neighbor sessions with a neighbor device using MD5 authentication is to start sending MD5 packets when the physical interface is up.

The IP-RIP Delay Start feature is often used when a Cisco device is configured to establish a RIPv2 neighbor relationship using MD5 authentication with a non-Cisco device over a Frame Relay network. When RIPv2 neighbors are connected over Frame Relay, it is possible for the serial interface connected to the Frame Relay network to be up while the underlying Frame Relay circuits are not yet ready to transmit and receive data.

When a serial interface is up and the Frame Relay circuits are not yet operational, any MD5 packets that the device attempts to transmit over the serial interface are dropped. When MD5 packets are dropped because the Frame Relay circuits over which the packets need to be transmitted are not yet operational, the sequence number of the first MD5 packet received by the neighbor device after the Frame Relay circuits become active will be greater than 0. Some non-Cisco devices will not allow an MD5-authenticated RIPv2 neighbor session to start when the sequence number of the first MD5 packet received from the other device is greater than 0.

The differences in vendor implementations of MD5 authentication for RIPv2 are probably a result of the ambiguity of the relevant RFC (RFC 2082) with respect to packet loss. RFC 2082 suggests that devices should be ready to accept either a sequence number of 0 or a sequence number higher than the last sequence number received. For more information about MD5 message reception for RIPv2, see section 3.2.2 of RFC 2082 at the following url: http://www.ietf.org/rfc/rfc2082.txt.

The IP-RIP Delay Start feature is supported over other interface types such as Fast Ethernet and Gigabit Ethernet.

Cisco devices allow an MD5-authenticated RIPv2 neighbor session to start when the sequence number of the first MD5 packet received from the other device is greater than 0. If you are using only Cisco devices in your network, you do not need to use the IP-RIP Delay Start feature.

Offset-list

An offset list is the mechanism for increasing incoming and outgoing metrics to routes learned via RIP. This is done to provide a local mechanism for increasing the value of routing metrics. Optionally, you can limit the offset list with either an access list or an interface.

Timers

Routing protocols use several timers that determine such variables as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs. You can make the following timer adjustments:

- The rate (time in seconds between updates) at which routing updates are sent
- The interval of time (in seconds) after which a route is declared invalid
- The interval (in seconds) during which routing information regarding better paths is suppressed
- The amount of time (in seconds) that must pass before a route is removed from the routing table
- The amount of time for which routing updates will be postponed

It also is possible to tune the IP routing support in the software to enable faster convergence of the various IP routing algorithms, and, hence, quicker fallback to redundant routers. The total effect is to minimize disruptions to end users of the network in situations where quick recovery is essential.

How to Configure RIP

Enabling RIP and Configuring RIP Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **network** *ip-address*
5. **neighbor** *ip-address*
6. **offset-list** [*access-list-number* | access-list-*name*] {**in** | **out**} *offset* [*interface-type interface-number*]
7. **timers basic** *update invalid holddown flush* [*sleeptime*]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router rip** **Example:** `Device(config)# router rip`	Enables a RIP routing process and enters router configuration mode.
Step 4	**network** *ip-address* **Example:** `Device(config-router)# network 10.1.1.0`	Associates a network with a RIP routing process.
Step 5	**neighbor** *ip-address* **Example:** `Device(config-router)# neighbor 10.1.1.2`	Defines a neighboring device with which to exchange routing information.
Step 6	**offset-list** [*access-list-number* | access-list-*name*] {**in** | **out**} *offset* [*interface-type interface-number*]	(Optional) Applies an offset list to routing metrics.

	Command or Action	Purpose
	Example: `Device(config-router)# offset-list 98 in 1 Ethernet 1/0`	
Step 7	**timers basic** *update invalid holddown flush* [*sleeptime*] **Example:** `Device(config-router)# timers basic 1 2 3 4`	(Optional) Adjusts routing protocol timers.
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Specifying a RIP Version and Enabling Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **version** {**1** | **2**}
5. **exit**
6. **interface** *type number*
7. **ip rip send version [1] [2]**
8. **ip rip receive version [1] [2]**
9. **ip rip authentication key-chain** *name-of-chain*
10. **ip rip authentication mode** {**text** | **md5**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router rip** **Example:** `Device(config)# router rip`	Enters router configuration mode.
Step 4	**version** {**1** | **2**} **Example:** `Device(config-router)# version 2`	Enables the Cisco software to send only RIP Version 2 (RIPv2) packets.
Step 5	**exit** **Example:** `Device(config-router)# exit`	Exits the router configuration mode and enters the global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface Ethernet 3/0`	Specifies an interface and enters interface configuration mode.
Step 7	**ip rip send version [1] [2]** **Example:** `Device(config-if)# ip rip send version 2`	Configures an interface to send only RIPv2 packets.
Step 8	**ip rip receive version [1] [2]** **Example:** `Device(config-if)# ip rip receive version 2`	Configures an interface to accept only RIPv2 packets.
Step 9	**ip rip authentication key-chain** *name-of-chain* **Example:** `Device(config-if)# ip rip authentication key-chain chainname`	Enables RIP authentication.
Step 10	**ip rip authentication mode** {**text** | **md5**} **Example:** `Device(config-if)# ip rip authentication mode md5`	Configures the interface to use message digest algorithm 5 (MD5) authentication (or let it default to plain-text authentication).
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Summarizing RIP Routes

RIP Version 2 supports automatic route summarization by default. The software summarizes subprefixes to the classful network boundary when classful network boundaries are crossed. If you have disconnected subnets, disable automatic route summarization to advertise the subnets. When route summarization is disabled, the software sends subnet and host routing information across classful network boundaries. To disable automatic summarization, use the **no auto-summary** command in router configuration mode.

Note Supernet advertisement (advertising any network prefix less than its classful major network) is not allowed in RIP route summarization, other than advertising a supernet learned in the routing tables. Supernets learned on any interface that is subject to configuration are still learned. For example, the following summarization is invalid: (invalid supernet summarization)

```
Router(config)# interface Ethernet 1
Router(config-if)# ip summary-address rip 10.0.0.0 252.0.0.0
.
.
>
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip summary-address rip** *ip-address network-mask*
5. **exit**
6. **router rip**
7. **no auto-summary**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters the interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface Ethernet 3/0`	
Step 4	**ip summary-address rip** *ip-address network-mask* **Example:** `Router(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0`	Specifies the IP address and network mask that identify the routes to be summarized.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits the interface configuration mode.
Step 6	**router rip** **Example:** `Router(config)# router rip`	Enters the router configuration mode.
Step 7	**no auto-summary** **Example:** `Router(config-router)# no auto-summary`	Used in router configuration mode, disables automatic summarization.
Step 8	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Enabling or Disabling Split Horizon

To enable or disable split horizon, use the following commands in interface configuration mode, as needed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip split-horizon**
5. **no ip split-horizon**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 3/0`	Enters interface configuration mode.
Step 4	**ip split-horizon** **Example:** `Router(config-if)# ip split-horizon`	Enables split horizon.
Step 5	**no ip split-horizon** **Example:** `Router(config-if)# no ip split-horizon`	Disables split horizon.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling the Validation of Source IP Addresses

Perform this task to disable the default function that validates the source IP addresses of incoming routing updates.

Note Split horizon for Frame Relay and SMDS encapsulation is disabled by default. Split horizon is not disabled by default for interfaces using any of the X.25 encapsulations. For all other encapsulations, split horizon is enabled by default.

In general, changing the state of the default is not recommended unless you are certain that your application requires making a change in order to advertise routes properly. Remember that if split horizon is disabled on a serial interface (and that interface is attached to a packet-switched network), you *must* disable split horizon for all routers in any relevant multicast groups on that network.

Note Summarized network will not be advertised when split horizon is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip split-horizon**
5. **exit**
6. **router rip**
7. **no validate-update-source**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 3/0`	Enters interface configuration mode.
Step 4	**ip split-horizon** **Example:** `Router(config-if)# ip split-horizon`	Enables split horizon.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 6	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode.
Step 7	**no validate-update-source** **Example:** `Router(config-router)# no validate-update-source`	Disables the validation of the source IP address of incoming RIP routing updates.

	Command or Action	Purpose
Step 8	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Interpacket Delay

Perform this to configure interpacket delay.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **exit**
5. **router rip**
6. **output-delay** *milliseconds*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 3/0`	Enters interface configuration mode.
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 5	**router rip** **Example:**	Enters router configuration mode.

	Command or Action	Purpose
	`Router(config)# router rip`	
Step 6	**output-delay** *milliseconds* **Example:** `Router(config-router)# output-delay 8`	Configures interpacket delay for outbound RIP updates.
Step 7	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Optimizing RIP over WAN

There are two problems when RIP is not optimized:

- Periodic broadcasting by RIP generally prevents WAN circuits from being closed.
- Even on fixed, point-to-point links, the overhead of periodic RIP transmissions could seriously interrupt normal data transfer because of the quantity of information that passes through the line every 30 seconds.

To overcome these limitations, triggered extensions to RIP cause RIP to send information on the WAN only when there has been an update to the routing database. Periodic update packets are suppressed over the interface on which this feature is enabled. RIP routing traffic is reduced on point-to-point, serial interfaces. Therefore, you can save money on an on-demand circuit for which you are charged for usage. Triggered extensions to RIP partially support RFC 2091, *Triggered Extensions to RIP to Support Demand Circuits* .

Perform the following task to enable triggered extensions to RIP and to display the contents of the RIP private database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *controller-number*
4. **ip rip triggered**
5. **end**
6. **show ip rip database** [*prefix mask*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *controller-number* **Example:** `Router(config)# interface serial3/0`	Configures a serial interface.
Step 4	**ip rip triggered** **Example:** `Router(config-if)# ip rip triggered`	Enables triggered extensions to RIP.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip rip database** [*prefix mask*] **Example:** `Router# show ip rip database`	Displays the contents of the RIP private database.

Configuring IP-RIP Delay Start for Routers Connected by a Frame Relay Network

The tasks in this section explain how to configure a router to use the IP-RIP Delay Start feature on a Frame Relay interface.

Timesaver Cisco routers allow an MD5-authenticated RIPv2 neighbor session to start when the sequence number of the first MD5 packet received from the other router is greater than 0. If you are using only Cisco routers in your network, you do not need to use the IP-RIP Delay Start feature.

Prerequisites

Your router must be running Cisco IOS Release 12.4(12) or a later release.

Note The IP-RIP Delay Start feature is supported over other interface types such as Fast Ethernet and Gigabit Ethernet. If your Cisco router cannot establish RIPv2 neighbor sessions using MD5 authentication with a non-Cisco device, the IP-RIP Delay Start feature might resolve the problem.

Restrictions

The IP-RIP Delay Start feature is required only when your Cisco router is configured to establish a RIPv2 neighbor relationship with a non-Cisco device and you want to use MD5 neighbor authentication.

Configuring RIPv2

This required task configures RIPv2 on the router.

This task provides instructions for only one of the many possible permutations for configuring RIPv2 on your router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **network** *ip-network*
5. **version** {**1** | **2**}
6. [**no**] **auto-summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router rip** **Example:** `Router(config)# router rip`	Enables a RIP routing process, which places you in router configuration mode.
Step 4	**network** *ip-network* **Example:** `Router(config-router)# network 192.168.0.0`	Associates a network with a RIP routing process.
Step 5	**version** {**1**	**2**} **Example:** `Router (config-router)# version 2`	Configures the software to receive and send only RIP Version 1 or only RIP Version 2 packets.

	Command or Action	Purpose
Step 6	[**no**] **auto-summary** **Example:** `Router(config-router)# no auto-summary`	Disables or restores the default behavior of automatic summarization of subnet routes into network-level routes.

Configuring Frame Relay on a Serial Subinterface

This required task configures a serial subinterface for Frame Relay.

Note This task provides instructions for only one of the many possible permutations for configuring Frame Relay on a subinterface. For more information about and instructions for configuring Frame Relay, see the Configuring Frame Relay part of the *Cisco IOS Wide-Area Networking Configuration Guide* .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address**
5. **encapsulation frame-relay [mfr number | ietf]**
6. **frame-relay lmi-type** {**cisco** | **ansi** | **q933a**}
7. **exit**
8. **interface** *type number/subinterface-number* {**point-to-point** | **multipoint**}
9. **frame-relay interface-dlci** *dlci* [**ietf** | **cisco**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface serial3/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Removes a previously configured IP address from the interface.
Step 5	**encapsulation frame-relay [mfr number | ietf]** **Example:** `Router(config-if)# encapsulation frame-relay ietf`	Specifies the type of Frame Relay encapsulation for the interface.
Step 6	**frame-relay lmi-type** {**cisco** | **ansi** | **q933a**} **Example:** `Router(config-if)# frame-relay lmi-type ansi`	Specifies the type of Frame Relay local management interface (LMI) for the interface.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 8	**interface** *type number/subinterface-number* {**point-to-point** | **multipoint**} **Example:** `Router(config)# interface serial3/0.1 point-to-point`	Specifies a subinterface and the connection type for the subinterface and enters subinterface configuration mode.
Step 9	**frame-relay interface-dlci** *dlci* [**ietf** | **cisco**] **Example:** `Router(config-subif)# frame-relay interface-dlci 100 ietf`	Assigns a data-link connection identifier (DLCI) to a Frame Relay subinterface.

Configuring IP with MD5 Authentication for RIPv2 and IP-RIP Delay on a Frame Relay Subinterface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *number*
5. **key-string** *string*
6. **exit**
7. **exit**
8. **interface** *type number*
9. **no cdp enable**

10. **ip address** *ip-address subnet-mask*
11. **ip rip authentication mode** {**text** | **md5**}
12. **ip rip authentication key-chain** *name-of-chain*
13. **ip rip initial-delay** *delay*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain rip-md5`	Specifies the name of a key chain and enters key chain configuration mode.
Step 4	**key** *number* **Example:** `Device(config-keychain)# key 123456`	Specifies the key identifier and enters key chain key configuration mode. The range is from 0 to 2147483647.
Step 5	**key-string** *string* **Example:** `Device(config-keychain-key)# key-string abcde`	Configures the key string.
Step 6	**exit** **Example:** `Device(config-keychain-key)# exit`	Exits key chain key configuration mode.
Step 7	**exit** **Example:** `Device(config-keychain)# exit`	Exits key chain configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface serial 3/0.1`	Specifies a subinterface and enters subinterface configuration mode.

	Command or Action	Purpose
Step 9	**no cdp enable** **Example:** `Device(config-subif)# no cdp enable`	Disables Cisco Discovery Protocol options on the interface. **Note** Cisco Discovery Protocol is not supported by non-Cisco devices; and the IP-RIP Delay Start feature is required only when you are connecting to a non-Cisco device. Therefore, you should disable Cisco Discovery Protocol on any interfaces on which you want to configure the IP-RIP Delay Start feature.
Step 10	**ip address** *ip-address subnet-mask* **Example:** `Device(config-subif)# ip address 172.16.10.1 255.255.255.0`	Configures an IP address for the Frame Relay subinterface.
Step 11	**ip rip authentication mode** {**text** | **md5**} **Example:** `Device(config-subif)# ip rip authentication mode md5`	Specifies the mode for RIPv2 authentication.
Step 12	**ip rip authentication key-chain** *name-of-chain* **Example:** `Device (config-subif)# ip rip authentication key-chain rip-md5`	Specifies a previously configured key chain for Routing Information Protocol Version (RIPv2) message digest algorithm 5 (MD5) authentication.
Step 13	**ip rip initial-delay** *delay* **Example:** `Device(config-subif)# ip rip initial-delay 45`	Configures the IP-RIP Delay Start feature on the interface. The device will delay sending the first MD5 authentication packet to the RIPv2 neighbor for the number of seconds specified by the *delay* argument. The range is from 0 to 1800.
Step 14	**end** **Example:** `Device(config-subif)# end`	Exits the subinterface configuration mode and returns to privileged EXEC mode.

Configuration Examples for RIP

Route Summarization Example

The following example shows how the **ip summary-address rip**router configuration command can be used to configure summarization on an interface. In this example, the subnets 10.1.3.0/25, 10.1.3.128/25, 10.2.1.0/24,

10.2.2.0/24, 10.1.2.0/24 and 10.1.1.0/24 can be summarized as shown below while sending the updates over an interface.

```
Router(config)#interface GigabitEthernet 0/2
Router(config-if)#ip summary-address rip 10.1.0.0 255.255.0.0
Router(config-if)#ip summary-address rip 10.2.0.0 255.255.0.0
Router(config-if)#ip summary-address rip 10.3.0.0 255.255.0.0
```

Split Horizon Examples

Two examples of configuring split horizon are provided.

Example 1

The following configuration shows a simple example of disabling split horizon on a serial link. In this example, the serial link is connected to an X.25 network.

```
Router(config)# interface Serial 0
Router(config-if)# encapsulation x25

Router(config-if)# no ip split-horizon
```

Example 2

In the next example, the figure below illustrates a typical situation in which the **no ip split-horizon** interface configuration command would be useful. This figure depicts two IP subnets that are both accessible via a serial interface on Router C (connected to a Frame Relay network). In this example, the serial interface on Router C accommodates one of the subnets via the assignment of a secondary IP address.

The Ethernet interfaces for Router A, Router B, and Router C (connected to IP networks 10.13.50.0, 10.155.120.0, and 10.20.40.0, respectively all have split horizon enabled by default, while the serial interfaces connected to networks 172.16.1.0 and 192.168.1.0 all have split horizon disabled with the **no ip split-horizon** command.The figure below shows the topology and interfaces.

In this example, split horizon is disabled on all serial interfaces. Split horizon must be disabled on Router C in order for network 172.16.0.0 to be advertised into network 192.168.0.0 and vice versa. These subnets overlap at Router C, interface S0. If split horizon were enabled on serial interface S0, it would not advertise a route back into the Frame Relay network for either of these networks.

Configuration for Router A

```
interface ethernet 1
 ip address 10.13.50.1
!
interface serial 1
 ip address 172.16.2.2
 encapsulation frame-relay
 no ip split-horizon
```

Configuration for Router B

```
interface ethernet 2
 ip address 10.155.120.1
```

```
!
interface serial 2
 ip address 192.168.1.2
 encapsulation frame-relay
 no ip split-horizon
```

Configuration for Router C

```
interface ethernet 0
 ip address 10.20.40.1
!
interface serial 0
 ip address 172.16.1.1
 ip address 192.168.1.1 secondary
 encapsulation frame-relay
 no ip split-horizon
```

Address Family Timers Example

The following example shows how to adjust individual address family timers. Note that the address family "notusingtimers" will use the system defaults of 30, 180, 180, and 240 even though timer values of 5, 10, 15, and 20 are used under the general RIP configuration. Address family timers are not inherited from the general RIP configuration.

```
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# timers basic 5 10 15 20
Router(config-router)# redistribute connected
Router(config-router)# network 5.0.0.0
Router(config-router)# default-metric 10
Router(config-router)# no auto-summary
Router(config-router)#
Router(config-router)# address-family ipv4 vrf abc
Router(config-router-af)# timers basic 10 20 20 20
Router(config-router-af)# redistribute connected
Router(config-router-af)# network 10.0.0.0
Router(config-router-af)# default-metric 5
Router(config-router-af)# no auto-summary
Router(config-router-af)# version 2
Router(config-router-af)# exit-address-family
Router(config-router)#
Router(config-router)# address-family ipv4 vrf xyz
Router(config-router-af)# timers basic 20 40 60 80
Router(config-router-af)# redistribute connected
Router(config-router-af)# network 20.0.0.0
Router(config-router-af)# default-metric 2
Router(config-router-af)# no auto-summary
Router(config-router-af)# version 2
Router(config-router-af)# exit-address-family
Router(config-router)#
Router(config-router)# address-family ipv4 vrf notusingtimers

Router(config-router-af)# redistribute connected
Router(config-router-af)# network 20.0.0.0
Router(config-router-af)# default-metric 2
Router(config-router-af)# no auto-summary
Router(config-router-af)# version 2
Router(config-router-af)# exit-address-family
Router(config-router)#
```

Example: IP-RIP Delay Start on a Frame Relay Interface

Additional References

The following sections provide references related to configuring Routing Information Protocol.

Related Documents

Related Topic	Document Title
Protocol-independent features, filtering RIP information, key management (available in RIP Version 2), and VLSM	*Configuring IP Routing Protocol-Independent Features*
IPv6 Routing: RIP for IPv6	*Cisco IOS IP Routing: RIP Configuration Guide*
RIP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: RIP Command Reference*
Configuring Frame Relay	*Cisco IOS Wide-Area Networking Configuration Guide*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
No new or modified MIBS are supported and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1058	*Routing Information Protocol*
RFC 2082	RIP-2 MD5 Authentication
RFC 2091	*Triggered Extensions to RIP to Support Demand Circuits*
RFC 2453	RIP version 2

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring RIP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 302: Feature Information for Configuring Routing Information Protocol

Feature Name	Releases	Feature Information
IP-RIP Delay Start	12.4(12), 15.0(1)M, 12.2(33)SRE, 15.0(1)SY	The IP-RIP Delay Start feature is used on Cisco routers to delay the initiation of RIPv2 neighbor sessions until the network connectivity between the neighbor routers is fully operational, thereby ensuring that the sequence number of the first MD5 packet that the router sends to the non-Cisco neighbor router is 0. The default behavior for a router configured to establish RIPv2 neighbor sessions with a neighbor router using MD5 authentication is to start sending MD5 packets when the physical interface is up. The following commands were introduced or modified: **ip rip initial-delay**.

Feature Name	Releases	Feature Information
IP Summary Address for RIPv2	12.0(7)T 12.1(3)T 12.1(14) 12.2(2)T 12.2(27)SBB 15.0(1)M 12.2(33)SRE 15.0S	The IP Summary Adddress for RIPv2 feature introduced the ability to summarize routes. Summarizing routes in RIP Version 2 improves scalability and efficiency in large networks. Summarizing IP addresses means that there is no entry for child routes (routes that are created for any combination of the individual IP addresses contained within a summary address) in the RIP routing table, reducing the size of the table and allowing the router to handle more routes. The following commands were introduced or modified by this feature: **ip summary-address rip**.
Routing Information Protocol	12.2(27)SBB 15.0(1)M 12.2(33)SRE 15.0S	Routing Information Protocol (RIP) is a commonly used routing protocol in small to medium TCP/IP networks. It is a stable protocol that uses a distance-vector algorithm to calculate routes.
Triggered RIP	12.0(1)T 15.0(1)M 12.2(33)SRE 15.0S	Triggered RIP was introduced to overcome constant RIP updates over expensive circuit-based WAN links. Triggered extensions to RIP cause RIP to send information on the WAN only when there has been an update to the routing database. Periodic update packets are suppressed over the interface on which this feature is enabled. RIP routing traffic is reduced on point-to-point, serial interfaces. The following commands were introduced or modified: **ip rip triggered**, **show ip rip database**.

Glossary

address family --A group of network protocols that share a common format of network address. Address families are defined by RFC 1700.

IS-IS --Intermediate System-to-Intermediate System. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, where routers exchange routing information based on a single metric, to determine network topology.

RIP --Routing Information Protocol.RIP is a dynamic routing protocol used in local and wide area networks.

VRF --VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

CHAPTER 249

BFD for RIPv2 Support

The BFD for RIPv2 Support feature is used to facilitate an alternate path selection when a neighboring router is inactive.

Routing Information Protocol (RIP) uses the timeout of prefixes of a particular neighbor to identify if a neighbor is inactive. By default, the timeout is 180 seconds; that is, although the next-hop router is inactive, the RIP router will still broadcast prefixes for up to 180 seconds.

Bidirectional Forward Detection (BFD) is a protocol that provides subsecond failure detection using a single, common standardized mechanism that is independent of media and routing protocols.

Prerequisites for BFD for RIPv2 Support

BFD is independent of RIPv2 and must be enabled and functional on the router.

How to Configure BFD for RIPv2 Support Feature

Configuring BFD on RIPv2 Neighbors

Perform this task to configure BFD on RIPv2 neighbors:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **bfd all-interfaces**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router rip** **Example:** `Router(config)# router rip`	Configures the RIP routing process and enters router configuration mode.
Step 4	**bfd all-interfaces** **Example:** `Router(config-router)# bfd all-interfaces`	Enables BFD on all interfaces associated with the routing process. • RIPv2 registers with BFD and creates sessions for the neighbor when RIP updates are received. New neighbors are automatically enabled for BFD when the update packets are received. **Note** Alternatively, you can use the **neighbor** *ip-address* **bfd**command to enable BFD for a specific RIP neighbor.
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to global configuration mode.

Configuration Example for BFD for RIPv2 Support Feature

Example Configuring BFD for a RIPv2 Neighbor

The following example shows how to configure BFD for all interfaces associated with a RIPv2 neighbor:

```
!
interface GigabitEthernet 0/0/0
 ip address 10.10.10.1 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 5
end
!
interface GigabitEthernet 0/0/1
```

```
 ip address 10.10.20.1 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 5
end
!
router rip
 version 2
 redistribute connected
 network 10.0.0.0
 neighbor 10.10.20.2 bfd
 bfd all-interfaces
 no auto-summary
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS IP Routing: Protocol-Independent Commands	*Cisco IOS IP Routing: Protocol-Independent Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	--

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for BFD for RIPv2 Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 303: Feature Information for BFD for RIPv2 Support

Feature Name	Releases	Feature Information
BFD for RIPv2 Support	Cisco IOS XE Release 3.3	The BFD for RIPv2 Support feature is used to facilitate alternate path selection when a neighboring router is inactive. The following commands were introduced or modified: **bfd all-interfaces, debug ip rip bfd events, neighbor (RIP),**and **show ip rip neighbor**.

CHAPTER 250

IPv6: RIPng VRF-Aware Support

The IPv6: RIPng VRF-Aware Support feature uses separate routing tables for every provider edge-customer edge (PE-CE) scenario, thus allowing improved route protection, modularity, and a potential reduction in the size of the routing table.

Information About IPv6: RIPng VRF-Aware Support

IPv6 Routing: RIP for IPv6

IPv6 Routing Information Protocol (RIP) functions the same and offers the same benefits as IPv4 RIP. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes and the use of the all-RIP-devices multicast group address, FF02::9, as the destination address for RIP update messages.

IPv6: RIPng VRF-Aware Support

When not in Virtual Routing and Forwarding (VRF) mode, every IPv6 Routing Information Protocol (RIP)—also know as RIP Next Generation (RIPng)—process and the configuration associated with it, keeps all the routes in the same routing table. In other routing protocols, it is often required to keep the protocol-related routes stored in separate routing tables.

The IPv6: RIPng VRF-Aware Support feature enables isolation, modularity, and potential performance improvement by reducing the number of routes stored in a single routing table. It also allows a network administrator to create different RIP routing tables and share the same protocol configuration stored in a single RIP protocol configuration block.

How to Configure IPv6: RIPng VRF-Aware Support

Configuring IPv6: RIPng VRF-Aware Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **vrf definition** *vrf-name*
5. **address-family ipv6**
6. **exit**
7. **exit**
8. **ipv6 rip vrf-mode enable**
9. **ipv6 router rip** *rip-process-name*
10. **exit**
11. **interface** *type number*
12. **vrf forwarding** *vrf-name*
13. **ipv6 enable**
14. **ipv6 rip** *rip-process-name* **enable**
15. **end**
16. **debug ipv6 rip vrf** *vrf-name*
17. **show ipv6 rip vrf** *vrf-name* **next-hops**
18. **show ipv6 rip vrf** *vrf-name* **database**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device # configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device (config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**vrf definition** *vrf-name* **Example:**	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.

	Command or Action	Purpose
	`Device(config)# vrf definition vrf1`	
Step 5	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and enables IPv6 address prefixes.
Step 6	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 8	**ipv6 rip vrf-mode enable** **Example:** `Device (config)# ipv6 rip vrf-mode enable`	Enables VRF support for IPv6 RIP routing and enters RTR entry configuration mode.
Step 9	**ipv6 router rip** *rip-process-name* **Example:** `Device (config)# ipv6 router rip myrip`	Creates an IPv6 Routing Information Protocol (RIP) routing process instance.
Step 10	**exit** **Example:** `Device (config-rtr)# exit`	Exits RTR entry configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device (config)# interface Ethernet 0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 12	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Binds the interface to the specified VRF routing instance table and removes all the Layer 3 interface configuration that is available when the command is entered.
Step 13	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface.
Step 14	**ipv6 rip** *rip-process-name* **enable** **Example:** `Device(config-if)# ipv6 rip myrip enable`	Enables an IPv6 RIP routing process on the interface.
Step 15	**end** **Example:** `Device (config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 16	**debug ipv6 rip vrf** *vrf-name* **Example:** `Device# debug ipv6 rip vrf vrf1`	Displays debugging information related to VRF support for the specified IPv6 RIP VRF routing table instance.
Step 17	**show ipv6 rip vrf** *vrf-name* **next-hops** **Example:** `Device# show ipv6 rip vrf vrf1 next-hops`	Displays the next hops in the specified VRF RIPng routing table.
Step 18	**show ipv6 rip vrf** *vrf-name* **database** **Example:** `Device# show ipv6 rip vrf vrf1 database`	Displays the associated RIP local routing information base (RIB).

Configuration Examples for IPv6: RIPng VRF-Aware Support

Example: Configuring IPv6: RIPng VRF-Aware Support

```
Device> enable
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# vrf definition vrf1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# ipv6 rip vrf-mode enable
Device(config)# ipv6 router rip myrip
Device(config-rtr)# exit
Device(config)# interface Ethernet 0/0
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 rip myrip enable
Device(config-if)# end
```

Example: Verifying IPv6: RIPng VRF-Aware Support

```
Device> debug ipv6 rip vrf vrf1

RIP Routing Protocol debugging is on for vrf vrf1
Sending:
*Mar 15 11:23:08.508: RIPng: Sending multicast update on Ethernet0/0 for vrf for vrf vrf1
*Mar 15 11:23:08.508: src=2001:DB8:0:1:FFFF:1234::5
*Mar 15 11:23:08.508: dst=2001:DB8:0:1::1 (Ethernet0/0)
*Mar 15 11:23:08.508: sport=521, dport=521, length=52
*Mar 15 11:23:08.508: command=2, version=1, mbz=0, #rte=2
*Mar 15 11:23:08.508: tag=0, metric=1, prefix=6000::/64
*Mar 15 11:23:08.508: tag=0, metric=1, prefix=2000::/64
```

```
*Mar 15 11:23:08.508: RIPng: Packet waiting
*Mar 15 11:23:08.508: RIPng: Process vrf received own response on Loopback1
Receiving
*Mar 15 11:23:20.316: RIPng: Packet waiting
*Mar 15 11:23:20.316: RIPng: response received from FE80::A8BB:CCFF:FE00:7C00 on Ethernet0/0
for vrf
*Mar 15 11:23:20.316: src=2001:DB8:0:1:FFFF:1234::4 (Ethernet0/0)
*Mar 15 11:23:20.316: dst=2001:DB8::1
*Mar 15 11:23:20.316: sport=521, dport=521, length=32
*Mar 15 11:23:20.316: command=2, version=1, mbz=0, #rte=1
*Mar 15 11:23:20.316: tag=0, metric=1, prefix=AAAA::/64
```

```
Device> show ipv6 rip vrf vrf1 database

RIP VRF "vrf1", local RIB
FE80::A8BB:CCFF:FE00:7C00/Ethernet0/0 [1 paths]
```

```
Device> show ipv6 rip vrf vrf1 next-hops

RIP VRF "vrf1", Next Hops
 AAAA::/64, metric 2, installed
Ethernet0/0/FE80::A8BB:CCFF:FE00:7C00, expires in 177 secs
```

Additional References for IPv6: RIPng VRF-Aware Support

Related Documents

Related Topic	Document Title
IP Routing: RIP commands	Cisco IOS IP Routing: RIP Command Reference
IPv6 Routing: RIP for IPv6	*Cisco IOS IP Routing: RIP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2080	*RIPng for IPv6*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for IPv6: RIPng VRF-Aware Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 304: IPv6: RIPng VRF-Aware Support

Feature Name	Releases	Feature Information
IPv6: RIPng VRF-Aware Support	15.3(3)M 15.2(1)SY	When not virtual routing and forwarding (VRF) aware, IPv6 Routing Information Protocol (RIP), also known as RIP Next Generation (RIPng), works only with routes that are available in the default global routing table. When operating in VRF mode, RIPng, creates a separate routing table for each VRF instance. The IPv6: RIPng VRF-Aware Support feature enables the availability of separate routing tables for every provider edge-customer edge (PE-CE) scenario, thus allowing improved route protection, modularity, and a potential reduction in the size of the routing table. The following commands were introduced or modified: **clear ipv6 rip**,**debug ipv6 rip** , **ipv6 rip vrf-mode enable**, and **show ipv6 rip**.

PART IX

Tunneling

CHAPTER 251

mGRE Tunnel Support over IPv6

The mGRE Tunnel Support over IPv6 feature enables service providers to deploy IPv6 in their core infrastructure.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About mGRE Tunnel Support over IPv6

mGRE Support over IPv6

Multiple sites of a Dynamic Multipoint Virtual Private Network (DMVPN) are interconnected by IPv6. A single logical multipoint generic routing encapsulation (mGRE) tunnel interface interconnects one VPN site to another. An IPv6 subnet connects a tunnel interface with other tunnel interfaces from various VPN sites. All tunnel interfaces connecting VPN sites act as hosts on the logical IPv6 subnet. This structure is referred to as the tunnel overlay network.

To enable service providers deploy IPv6 in their core infrastructure, mGRE tunnels over IPv6 are supported. DMVPN customers may run either IPv4 or IPv6 in their local networks, so the overlay endpoints can be either IPv4 or IPv6. For an IPv6 transport endpoint, the overlay endpoint can either be an IPv4 or IPv6 private network address.

GRE has a protocol field that identifies the passenger protocol. GRE tunnels allow Intermediate System-to-Intermediate System (IS-IS) or IPv6 to be specified as a passenger protocol, which allows both IS-IS and IPv6 traffic to run over the same tunnel. If GRE did not have a protocol field, it would be impossible to distinguish whether the tunnel was carrying IS-IS or IPv6 packets.

How to Configure mGRE Tunnel Support over IPv6

Configuring mGRE Tunnel Support over IPv6

Perform this task on the hub and spoke device of the multipoint generic routing encapsulation (mGRE) tunnel.

Before you begin

Create a Next Hop Resolution Protocol (NHRP) ID to configure on a multipoint generic routing encapsulation (mGRE) tunnel.

For more information on configuring NHRP, see the "How to Configure NHRP" topic in the *IP Addressing : NHRP Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface tunnel** *tunnel-number*
5. **description** *description-string*
6. **ipv6 address** *ip-address mask*
7. **ipv6 nhrp map mulitcast dynamic**
8. **ipv6 nhrp network-id** *network-id*
9. **ipv6 nhrp holdtime** *seconds*
10. **ipv6 nhrp nhs** *ipv6- nhs-address*
11. **tunnel source** *ip-address* | *ipv6-address* | *interface-type* | *interface-number*
12. **tunnel mode gre multipoint ipv6**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast datagrams.
Step 4	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode. • The *tunnel-number* argument specifies the number of tunnel interfaces that you can create or configure. There is no limit on the number of tunnel interfaces that you can configure.
Step 5	**description** *description-string* **Example:** `Device(config-if)# description DMVPN HUB`	Configures information specific to the interface.
Step 6	**ipv6 address** *ip-address mask* **Example:** `Device(config-if)# ipv6 address 2001:0DB8:0C18:2::300/64`	Specifies the IPv6 address and mask of the hub.
Step 7	**ipv6 nhrp map mulitcast dynamic** **Example:** `Device(config-if)# ipv6 nhrp map multicast dynamic`	Enables NHRP to initiate multipoint GRE tunnels to register their unicast NHRP mappings.
Step 8	**ipv6 nhrp network-id** *network-id* **Example:** `Device(config-if)# ipv6 nhrp network-id 100`	Configures NHRP on an interface. The IPv6 NHRP network-id is a unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 9	**ipv6 nhrp holdtime** *seconds* **Example:** `Device(config-if)# ipv6 nhrp holdtime 100`	Configures the time in seconds that NBMA addresses are advertised as valid in NHRP response.
Step 10	**ipv6 nhrp nhs** *ipv6- nhs-address* **Example:** `Device(config-if)# ipv6 nhrp nhs 1101:1::1`	Specifies IPv6 prefix of one or more NHRP servers.
Step 11	**tunnel source** *ip-address* | *ipv6-address* | *interface-type* | *interface-number* **Example:** `Device(config-if)# tunnel source ethernet 0`	Configures the source address of a tunnel interface.
Step 12	**tunnel mode gre multipoint ipv6** **Example:** `Device(config-if)# tunnel mode gre multipoint ipv6`	Sets the encapsulation mode of the tunnel to mGRE IPv6.

	Command or Action	Purpose
Step 13	**end** **Example:** `Device(config-if)# end`	Exits to global configuration mode.

What to do next

Verify the mGRE tunnel over IPv6.

Verifying mGRE Tunnel Support over IPv6

The **show** commands can be entered in any order.

Before you begin

Configure mGRE tunnel over IPv6.

SUMMARY STEPS

1. **show interface tunnel** *tunnel-interface*
2. **show tunnel endpoints tunnel** *tunnel-interface*
3. **show ipv6 traffic**

DETAILED STEPS

Step 1 **show interface tunnel** *tunnel-interface*

This command displays information about the tunnel.

Example:

```
Device# show interface tunnel 1

Tunnel1 is up, line protocol is down
Hardware is Tunnel
Description: DMVPN Spoke 1
MTU 1456 bytes, BW 100 Kbit/sec, DLY 50000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel linestate evaluation down - transport reg down
Tunnel source Ethernet1/0
Tunnel Subblocks:
src-track:
Tunnel1 source tracking subblock associated with Ethernet1/0
Set of tunnels with source Ethernet1/0, 1 member (includes iterators), on interface <OK>
Tunnel protocol/transport multi-GRE/IPv6
Key disabled, sequencing disabled
Checksumming of packets disabled
Tunnel TTL 255
Tunnel transport MTU 1456 bytes
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
```

```
Last input never, output never, output hang never
```

Step 2 **show tunnel endpoints tunnel** *tunnel-interface*

This command displays tunnel interface endpoints and verifies if the tunnel is created correctly.

Example:

```
Device# show tunnel endpoints tunnel 1

Tunnel 1 running in multi-GRE/IPv6 mode
Endpoint transport 1101:2::1 Refcount 3 Base 0x2B83A87F83D8 Create Time 00:22:05
overlay 1101:1::1 Refcount 2 Parent 0x2B83A87F83D8 Create Time 00:22:05
Tunnel Subblocks:
tunnel-nhrp-sb:
NHRP subblock has 1 entries
```

Step 3 **show ipv6 traffic**

This command displays statistics about IPv6 traffic on a tunnel.

Example:

```
Device# show ipv6 traffic

IPv6 statistics:
        Rcvd:  46 total, 34 local destination
               0 source-routed, 0 truncated
               0 format errors, 0 hop count exceeded
               0 bad header, 0 unknown option, 0 bad source
               0 unknown protocol, 0 not a router
               0 fragments, 0 total reassembled
               0 reassembly timeouts, 0 reassembly failures
        Sent:  54 generated, 0 forwarded
               0 fragmented into 0 fragments, 0 failed
               8 encapsulation failed, 0 no route, 0 too big
               0 RPF drops, 0 RPF suppressed drops
        Mcast: 22 received, 21 sent

ICMP statistics:
        Rcvd: 37 input, 0 checksum errors, 0 too short
              0 unknown info type, 0 unknown error type
              unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
                       0 sa policy, 0 reject route
              parameter: 0 error, 0 header, 0 option
              0 hopcount expired, 0 reassembly timeout,0 too big
              0 bad embedded ipv6
              10 echo request, 0 echo reply
              0 group query, 0 group report, 0 group reduce
              0 router solicit, 7 router advert, 0 redirects
              4 neighbor solicit, 6 neighbor advert
        Sent: 47 output, 0 rate-limited
              unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
                       0 sa policy, 0 reject route
              parameter: 0 error, 0 header, 0 option
              0 hopcount expired, 0 reassembly timeout,0 too big
              0 echo request, 10 echo reply
              0 group query, 0 group report, 0 group reduce
              3 router solicit, 7 router advert, 0 redirects
              6 neighbor solicit, 6 neighbor advert

UDP statistics:
```

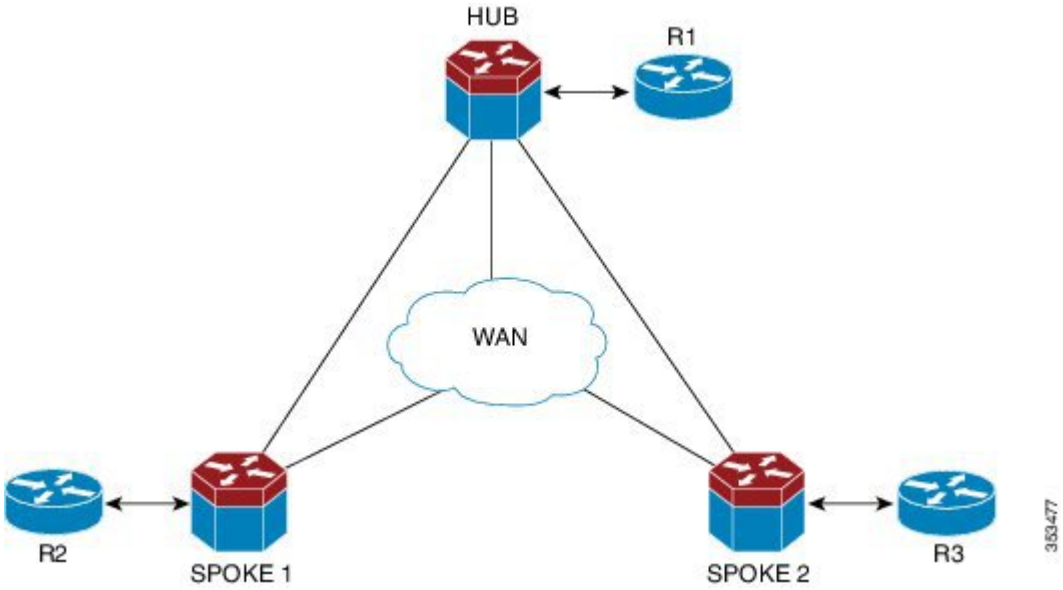
```
  Rcvd: 0 input, 0 checksum errors, 0 length errors
        0 no port, 0 dropped
  Sent: 0 output

TCP statistics:
  Rcvd: 0 input, 0 checksum errors
  Sent: 0 output, 0 retransmitted
```

Configuration Example for mGRE Tunnel over IPv6

Example for mGRE Tunnel over IPv6

mGRE Tunnel over IPv6

Configuring mGRE tunnel over IPv6 transport.




```
! Configure the topology

R1(config)# ipv6 unicast-routing
R1(config)# ipv6 cef
R1(config)# interface Ethernet0/1
R1(config-if)# ipv6 address 2001:DB8:1111:1111::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# ipv6 route ::/0 2001:DB8:1111:1111::2

! Configure the tunnel interface on hub

Hub(config)# ipv6 unicast-routing
Hub(config)# interface tunnel 1
Hub(config-if)# description DMVPN HUB
Hub(config-if)# ipv6 address 2001:DB8:1111:4444::1/64
```

```
Hub(config-if)# ipv6 nhrp map multicast dynamic
Hub(config-if)# ipv6 nhrp network-id 100
Hub(config-if)# ipv6 nhrp holdtime 100
Hub(config-if)# tunnel source Ethernet0/1
Hub(config-if)# tunnel mode gre multipoint ipv6

! Configure the physical interface on the hub

Hub(config)# ipv6 unicast-routing
Hub(config)# interface Ethernet0/0
Hub(config-if)# ipv6 address 2001:DB8:1111:2222::1/64
Hub(config-if)# no shutdown
Hub(config-if)# exit
Hub(config)# ipv6 route ::/0 2001:DB8:1111:2222::2


! Configure the tunnel interface on spoke

Spoke1(config)# ipv6 unicast-routing
Spoke1(config)# interface tunnel 1
Spoke1(config-if)# description DMVPN Spoke 1
Spoke1(config-if)# ipv6 address 2001:DB8:1111:4444::2/64
Spoke1(config-if)# ipv6 nhrp map multicast dynamic
Spoke1(config-if)# ipv6 nhrp map 2001:DB8:1111:4444::1/64 2001:DB8:1111:3333::1
Spoke1(config-if)# ipv6 nhrp map multicast 2001:DB8:1111:3333::1
Spoke1(config-if)# ipv6 nhrp network-id 100
Spoke1(config-if)# ipv6 nhrp holdtime 100
Spoke1(config-if)# ipv6 nhrp nhs 2001:DB8:1111:4444::1
Spoke1(config-if)# tunnel source Ethernet0/0
Spoke1(config-if)# tunnel mode gre multipoint ipv6

! Configure the physical interface on the spoke

Spoke1(config)# interface Ethernet0/0
Spoke1(config-if)# ipv6 address 2001:DB8:1111:2222::2/64
Spoke1(config-if)# exit

! Configure the R2 device at the spoke

R2(config)# interface Ethernet0/1
R2(config-if)# ipv6 address 2001:DB8:1111:3333::1/64
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# ipv6 route 2001:DB8:1111:1111::/64 2001:DB8:1111:2222::1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: *http://www.cisco.com/go/mibs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	*http://www.cisco.com/cisco/web/support/index.html*

Feature Information for mGRE Tunnel Support over IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 305: Feature Information for mGRE Tunnel Support over IPv6

Feature Name	Releases	Feature Information
mGRE Tunnel Support over IPv6	15.2(1)T XE Release 3.8S	mGRE tunnels are configured to enable service providers deploy IPv6 in their core infrastructure.

CHAPTER 252

IP over IPv6 Tunnels

IPv6 supports IP over IPv6 tunnels, which includes the following:

- Generic routing encapsulation (GRE) IPv4 tunnel support for IPv6 traffic—IPv6 traffic can be carried over IPv4 GRE tunnels using the standard GRE tunneling technique that is designed to provide the services to implement any standard point-to-point encapsulation scheme. The primary use of GRE tunnels is for stable connections that require regular secure communication between two edge devices or between an edge device and an end system. The edge devices and the end systems must be dual-stack implementations.
- GRE support over IPv6 transport—GRE has a protocol field that identifies the passenger protocol. GRE tunnels allow Intermediate System-to-Intermediate System (IS-IS) or IPv6 to be specified as a passenger protocol, which allows both IS-IS and IPv6 traffic to run over the same tunnel.
- VRF-aware IPv4/IPv6 over IPv6 tunnels - Virtual Routing and Forwarding (VRF)-aware tunnels are used to connect customer networks separated by untrusted core networks or core networks with different infrastructures (IPv4 or IPv6).

Information About IP over IPv6 Tunnels

GRE IPv4 Tunnel Support for IPv6 Traffic

IPv6 traffic can be carried over IPv4 GRE tunnels using the standard GRE tunneling technique that is designed to provide the services to implement any standard point-to-point encapsulation scheme. As in IPv6 manually configured tunnels, GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol but, in this case, carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol.

The primary use of GRE tunnels is for stable connections that require regular secure communication between two edge devices or between an edge device and an end system. The edge devices and the end systems must be dual-stack implementations.

GRE Support over IPv6 Transport

GRE has a protocol field that identifies the passenger protocol. GRE tunnels allow Intermediate System-to-Intermediate System (IS-IS) or IPv6 to be specified as a passenger protocol, which allows both IS-IS and IPv6 traffic to run over the same tunnel. If GRE did not have a protocol field, it would be impossible to distinguish whether the tunnel was carrying IS-IS or IPv6 packets. The GRE protocol field makes it desirable to tunnel IS-IS and IPv6 inside GRE.

How to Configure IP over IPv6 Tunnels

The following tasks describe how to configure an IPv6 tunnel. IPv6 or IPv4 packets can be forwarded on this tunnel.

Configure CDP Over GRE IPv6 Tunnels

Perform this task to configure a GRE tunnel on an IPv6 network. GRE tunnels can be configured to run over an IPv6 network layer and transport IPv6 and IPv4 packets through IPv6 tunnels.

Note You must enable IPv6 or configure IPv6 MTU size more than 1500 on a tunnel's exit interface to avoid receiving warning messages.

Before you begin

When GRE IPv6 tunnels are configured, IPv6 addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 or IPv6 addresses. The host or device at each end of the configured tunnel must support both IPv4 and IPv6 protocol stacks.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**CDP enable** **Example:**	Enables Cisco Discovery Protocol on the interface.

	Command or Action	Purpose
	`Device(config)# CDP enable`	
Step 5	**tunnel source** {*ipv6-address* | *interface-type* |*interface-number* } **Example:** `Device(config-if)# tunnel source ethernet 0`	Specifies the source IPv6 address or the source interface type and number for the tunnel interface. • If an interface type and number are specified, the interface must be configured with an IPv6 address. **Note** For more information on the tunnel source command, refer to the IPv6 command reference guide.
Step 6	**tunnel destination** *ipv6-address* **Example:** `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	Specifies the destination IPv6 address for the tunnel interface. **Note** For more information on the tunnel destination command, refer to the IPv6 command reference guide.
Step 7	**tunnel mode gre ipv6** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel interface. Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for IP over IPv6 Tunnels

Example: IPv6 over IPv6 Tunnel

Example: Configuring CE1

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!
```

```
ipv6 route 2001:DB8:2:5::/64 2001:DB8:2:1::2
ipv6 route 2001:DB8:2:9::/64 2001:DB8:2:1::2
!
```

Example: Configuring PE1

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ipv6 address
 ipv6 address 2001:DB8:2:9::1/64
 tunnel source 2001:DB8:2:2::1
 tunnel mode ipv6
 tunnel destination 2001:DB8:2:4::2
 exit
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
!
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!

ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:2::2
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:2::2
ipv6 route 2001:DB8:2:5::/64 Tunnel0 2001:DB8:2:9::2
```

Example: Configuring PE2

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ipv6 address
 ipv6 address 2001:DB8:2:9::2/64
 tunnel source 2001:DB8:2:4::2
 tunnel mode ipv6
 tunnel destination 2001:DB8:2:2::1
 exit
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:5::1/64
 no shutdown
 exit
!
```

```
interface Ethernet0/1
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:4::1
ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:4::1
ipv6 route 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:9::1
```

Example: Configuring CE2

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:5::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:1::/64 2001:DB8:2:5::1
ipv6 route 2001:DB8:2:9::/64 2001:DB8:2:5::1
!
```

Example: Configuring Core Device 1

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet1/0
 no ipv6 address
 no shutdown
 ipv6 address 2001:DB8:2:3::1/64
 exit
!
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:3::2
```

Example: Configuring Core Device 2

```
!
ipv6 unicast-routing
ipv6 cef
```

```
!
interface Ethernet0/1
 no ip address
 ipv6 address 2001:DB8:2:4::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 no ip address
 ipv6 address 2001:DB8:2:3::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:3::1
```

Example: IPv4 over IPv6 Tunnel

Example: Configuring CE1

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ip address
 ip address 192.168.1.1 255.255.255.0
 no shutdown
 exit
!
ip route 192.168.5.0 255.255.255.0 192.168.1.2
ip route 192.168.9.0 255.255.255.0 192.168.1.2
!
```

Example: Configuring PE1

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ip address
 ip address 192.168.9.1 255.255.255.0
 tunnel source 2001:DB8:2:2::1
 tunnel destination 2001:DB8:2:4::2
 tunnel mode ipv6
 exit
!
interface Ethernet0/0
 no ip address
 ip address 192.168.1.2 255.255.255.0
 no shutdown
 exit
!
!
```

```
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!

ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:2::2
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:2::2
ip route 192.168.5.0 255.255.255.0 Tunnel 0 192.168.9.2
```

Example: Configuring PE2

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ip address
 ip address 192.168.9.2 255.255.255.0
 tunnel source 2001:DB8:2:4::2
 tunnel destination 2001:DB8:2:2::1
 tunnel mode ipv6
 exit
!
interface Ethernet0/0
 no ip address
 ip address 192.168.5.1 255.255.255.0
 no shutdown
 exit
!
interface Ethernet0/1
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:4::1
ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:4::1
ip route 192.168.1.0 255.255.255.0 Tunnel 0 192.168.9.1
```

Example: Configuring CE2

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ip address
 ip address 192.168.5.2 255.255.255.0
 no shutdown
 exit
!
ip route 192.168.1.0 255.255.255.0 192.168.1.2
```

```
ip route 192.168.9.0 255.255.255.0 192.168.1.2

!
```

Example: Configuring Core Device 1

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet1/0
 no ipv6 address
 no shutdown
 ipv6 address 2001:DB8:2:3::1/64
 exit
!
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:3::2
```

Example: Configuring Core Device 2

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/1
 no ip address
 ipv6 address 2001:DB8:2:4::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 no ip address
 ipv6 address 2001:DB8:2:3::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:3::1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFC*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP over IPv6 Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 306: Feature Information for IP over IPv6 Tunnels

Feature Name	Releases	Feature Information
IP over IPv6 Tunnels	12.2(30)S 12.2(33)SRA 12.3(7)T 12.4 12.4(2)T 15.0(1)S Cisco IOS XE Release 2.1 15.1(1)SY	IP over IPv6 Tunnels feature is supported. The following commands were introduced or modified: **tunnel destination**, **tunnel mode ipv6**, **tunnel mode gre ipv6**, **tunnel source**.

CHAPTER 253

Manually Configured IPv6 over IPv4 Tunnels

This feature provides support for manually configured IPv6 over IPv4 tunnels. A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone.

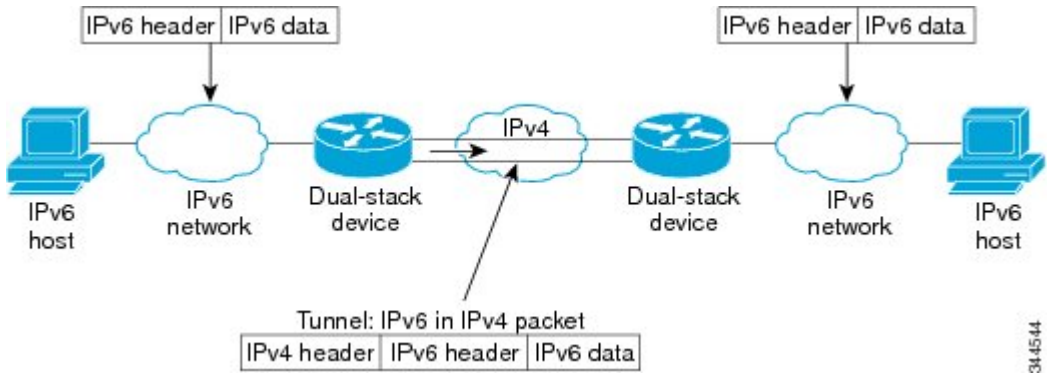
Information About Manually Configured IPv6 over IPv4 Tunnels

Overlay Tunnels for IPv6

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the figure below). By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border devices or between a border device and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks. IPv6 supports the following types of overlay tunneling mechanisms:

- Manual
- Generic routing encapsulation (GRE)
- IPv4-compatible
- 6to4
- Intrasite Automatic Tunnel Addressing Protocol (ISATAP)

Figure 213: Overlay Tunnels



Note Overlay tunnels reduce the maximum transmission unit (MTU) of an interface by 20 octets (assuming that the basic IPv4 packet header does not contain optional fields). A network that uses overlay tunnels is difficult to troubleshoot. Therefore, overlay tunnels that connect isolated IPv6 networks should not be considered a final IPv6 network architecture. The use of overlay tunnels should be considered as a transition technique toward a network that supports both the IPv4 and IPv6 protocol stacks or just the IPv6 protocol stack.

Use the table below to help you determine which type of tunnel that you want to configure to carry IPv6 packets over an IPv4 network.

Table 307: Suggested Usage of Tunnel Types to Carry IPv6 Packets over an IPv4 Network

Tunneling Type	Suggested Usage	Usage Notes
Manual	Simple point-to-point tunnels that can be used within a site or between sites.	Can carry IPv6 packets only.
GRE- and IPv4-compatible	Simple point-to-point tunnels that can be used within a site or between sites.	Can carry IPv6, Connectionless Network Service (CLNS), and many other types of packets.
IPv4- compatible	Point-to-multipoint tunnels.	Uses the ::/96 prefix. We do not recommend using this tunnel type.
6to4	Point-to-multipoint tunnels that can be used to connect isolated IPv6 sites.	Sites use addresses from the 2002::/16 prefix.
6RD	IPv6 service is provided to customers over an IPv4 network by using encapsulation of IPv6 in IPv4.	Prefixes can be from the SP's own address block.
ISATAP	Point-to-multipoint tunnels that can be used to connect systems within a site.	Sites can use any IPv6 unicast addresses.

Individual tunnel types are discussed in detail in this document. We recommend that you review and understand the information about the specific tunnel type that you want to implement. When you are familiar with the

type of tunnel you need, see the table below for a summary of the tunnel configuration parameters that you may find useful.

Table 308: Tunnel Configuration Parameters by Tunneling Type

Tunneling Type	Tunnel Configuration Parameter			
Tunnel Mode	**Tunnel Source**	**Tunnel Destination**	**Interface Prefix or Address**	
Manual	ipv6ip	An IPv4 address, or a reference to an interface on which IPv4 is configured.	An IPv4 address.	An IPv6 address.
GRE/IPv4	gre ip		An IPv4 address.	An IPv6 address.
IPv4- compatible	ipv6ip auto-tunnel		Not required. These are all point-to-multipoint tunneling types. The IPv4 destination address is calculated, on a per-packet basis, from the IPv6 destination.	Not required. The interface address is generated as ::*tunnel-source*/96.
6to4	ipv6ip 6to4			An IPv6 address. The prefix must embed the tunnel source IPv4 address.
6RD	ipv6ip 6rd			An IPv6 address.
ISATAP	ipv6ip isatap			An IPv6 prefix in modified eui-64 format. The IPv6 address is generated from the prefix and the tunnel source IPv4 address.

IPv6 Manually Configured Tunnels

A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The primary use is for stable connections that require regular secure communication between two edge devices or between an end system and an edge device, or for connection to remote IPv6 networks.

An IPv6 address is manually configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or device at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks. Manually configured tunnels can be configured between border devices or between a border device and a host. Cisco Express Forwarding switching can be used for IPv6 manually configured tunnels, or Cisco Express Forwarding switching can be disabled if process switching is needed.

How to Enable Manually Configured IPv6 over IPv4 Tunnels

Configuring Manual IPv6 Tunnels

Before you begin

With manually configured IPv6 tunnels, an IPv6 address is configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or device at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. Enter one of the following commands:
   - **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
   - **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *interface-type interface-number*}
6. **tunnel destination** *ip-address*
7. **tunnel mode ipv6ip**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	Enter one of the following commands: • **ipv6 address** {*ipv6-address*/*prefix-length*	*prefix-name sub-bits*/*prefix-length*} • **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] **Example:** `Device(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface. • If you specify the **eui-64** keyword, the software configures an IPv6 address for an interface and enables IPv6 processing on the interface using an EUI-64 interface ID in the low-order 64 bits of the address. **Note** See the "Implementing IPv6 Addressing and Basic Connectivity" module for more information on configuring IPv6 addresses.
Step 5	**tunnel source** {*ip-address*	*interface-type interface-number*} **Example:**	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. • If an interface is specified, the interface must be configured with an IPv4 address.

	Command or Action	Purpose
	`Device(config-if)# tunnel source gigabitethernet 0/0/0`	
Step 6	**tunnel destination** *ip-address* **Example:** `Device(config-if)# tunnel destination 192.168.30.1`	Specifies the destination IPv4 address or hostname for the tunnel interface.
Step 7	**tunnel mode ipv6ip** **Example:** `Device(config-if)# tunnel mode ipv6ip`	Specifies a manual IPv6 tunnel. **Note** The **tunnel mode ipv6ip** command specifies IPv6 as the passenger protocol and IPv4 as both the encapsulation and transport protocol for the manual IPv6 tunnel.
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for Manually Configured IPv6 over IPv4 Tunnels

Example: Configuring Manual IPv6 Tunnels

The following example configures a manual IPv6 tunnel between router A and router B. In the example, tunnel interface 0 for both router A and router B is manually configured with a global IPv6 address. The tunnel source and destination addresses are also manually configured.

Router A Configuration

```
interface ethernet 0
 ip address 192.168.99.1 255.255.255.0
interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::3/127
 tunnel source ethernet 0
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

Router B Configuration

```
interface ethernet 0
 ip address 192.168.30.1 255.255.255.0
interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::2/127
 tunnel source ethernet 0
```

```
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

Example: IPv6 over GRE IPv4 Tunnel

Example: Configuring CE1

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:1::2
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:1::2
!
```

Example: Configuring PE1

```
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ip address
 ipv6 address 2001:DB8:2:4::1/64
 tunnel source 10.22.22.22
 tunnel destination 10.44.44.44
 exit
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.22 255.255.255.0
 no shutdown
 exit
!
ip route 10.44.44.0 255.255.255.0 10.22.22.23
ipv6 route 2001:DB8:2:2::/64 Tunnel0 2001:DB8:2:4::2
```

Example: Configuring PE2

```
!
ipv6 unicast-routing
ipv6 cef
```

```
!
interface Tunnel0
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 tunnel source 10.44.44.44
 tunnel destination 10.22.22.22
 exit
!
interface Ethernet0/0 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.44 255.255.255.0
 no shutdown
 exit
!
ip route 10.22.22.0 255.255.255.0 10.44.44.43
!
ipv6 route 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:4::1
!
```

Example: Configuring CE2

```
!
ipv6 unicast-routing
ipv6 cef
!
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:1::/64 2001:DB8:2:2::1
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:2::1
!
```

Example: Configuring Device X

```
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.43 255.255.255.0
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.23 255.255.255.0
 no shutdown
 exit
!
```

Example: Verifying the Tunnel Configuration

From CE1

```
Device# ping ipv6 2001:db8:2:2::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/43 ms

Device# ping ipv6 2001:db8:2:2::2 source 2001:db8:2:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:2:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

From PE1

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source 10.22.22.22
   IP transport: output interface Ethernet1/1 next hop 10.22.22.23
   Application ID 1: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up
   Tunnel Source Flags: Local
   Transport IPv4 Header DF bit cleared
   OCE: IP tunnel decap
   Provider: interface Tu0, prot 47
     Performs protocol check [47]
     Protocol Handler: GRE: opt 0x0
       ptype: ipv4 [ipv4 dispatcher: punt]
       ptype: ipv6 [ipv6 dispatcher: from if Tu0]
       ptype: mpls [mpls dispatcher: drop]
       ptype: otv [mpls dispatcher: drop]
       ptype: generic [mpls dispatcher: drop]
There are 0 tunnels running over the EON IP protocol
There are 0 tunnels running over the IPinIP protocol
There are 0 tunnels running over the NOSIP protocol
There are 0 tunnels running over the IPv6inIP protocol
There are 0 tunnels running over the RBSCP/IP protocol

Device# show ip route 10.44.44.44

Routing entry for 10.44.44.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.22.22.23
      Route metric is 0, traffic share count is 1

Device# debug ipv6 icmp

  ICMP Packet debugging is on
*Jan  1 10:57:37.882: ICMPv6: Sent R-Advert, Src=FE80::A8BB:CCFF:FE00:5200, Dst=FF02::1
```

```
*Jan  1 11:00:18.634: ICMPv6: Received R-Advert, Src=FE80::A8BB:CCFF:FE00:5200,Dst=FF02::1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: *http://www.cisco.com/go/mibs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Manually Configured IPv6 over IPv4 Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 309: Feature Information for Manually Configured IPv6 over IPv4 Tunnels

Feature Name	Releases	Feature Information
IPv6 Tunneling: Manually Configured IPv6 over IPv4 Tunnels	Cisco IOS XE Release 2.1	A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The following commands were introduced or modified: **tunnel destination**, **tunnel ipv6ip**, **tunnel source**.

CHAPTER 254

Configuring Physical Interfaces

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Configuration Information

- For information about using the Gigabit Ethernet Management Ethernet interface, see the *Cisco Catalyst 8300 and 8200 Series Edge Platforms Software Configuration Guide* at:

https://www.cisco.com/c/en/us/td/docs/routers/cloud_edge/c8300/software_config/cat8300swcfg-xe-17-book.html

Command Reference Information

- Complete descriptions of the commands used to configure interfaces are included in the *Cisco IOS Interface and Hardware Component Command Reference* at:

http://www.cisco.com/en/US/docs/ios/interface/command/reference/ir_book.html

- For information about other Cisco IOS XE commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases* , at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html .

CHAPTER 255

Configuring Virtual Interfaces

Virtual interfaces are software-based interfaces that you create in the memory of the networking device using Cisco IOS XE commands. Virtual interfaces do not have a hardware component such as the RJ-45 female port on a 100BASE-T Fast Ethernet network interface card. This module describes the four common types of virtual, or logical, interfaces that can be configured using Cisco IOS XE software:

- Loopback interfaces
- Null interfaces
- Subinterfaces
- Tunnel interfaces

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Virtual Interfaces

Before virtual interfaces can be used in your network, you must have some physical (hardware) interfaces configured and you must be able to communicate between the networking devices on which you wish to use virtual interfaces.

Information About Configuring Virtual Interfaces

Virtual Interfaces

Virtual interfaces are network interfaces that are not associated with a physical interface. Physical interfaces have some form of physical element--for example, an RJ-45 male connector on an Ethernet cable. Virtual interfaces exist only in software; there are no physical elements. You identify an individual virtual interface using a numerical ID after the virtual interface name. For example: loopback 0, tunnel 1, and fastethernet 0/0/0.1. The ID is unique per virtual interface type to make the entire name string unique; for example both a loopback 0 interface and a null 0 interface can exist, but two loopback 0 interfaces cannot exist in a single networking device.

Cisco IOS XE software supports four types of virtual interfaces:

- Loopback
- Null
- Subinterface
- Tunnel

Benefits of Virtual Interfaces

A loopback interface can provide a stable interface on which you can assign a Layer 3 address such as an IP or IPX address. This address can be configured as the source address when the networking device needs to send data for protocols such as NetFlow or Cisco Discovery Protocol (CDP) to another device in your network and you want the receiving device to always see the same source IP address from the networking device. This is an issue in networks with multiple equal-cost paths because under normal circumstances the packets that are generated by a networking device use the IP address from the outbound interface as the source address for the packets and because in a network with two or more equal-cost paths from the networking device to the receiving host each packet might use a different outbound interface.

A null interface provides an alternative method of filtering without the overhead involved with using access lists. For example, instead of creating an outbound access list that prevents traffic to a destination network from being transmitted out an interface, you can configure a static route for the destination network that points to the null interface.

Subinterfaces were invented as a method of virtually subdividing a physical interface into two or more interfaces so that the IP routing protocols would see the network connection to each remote networking device as a separate physical interface even though the subinterfaces share a common physical interface. One of the first uses of subinterfaces was to resolve the problem with split horizon on Frame Relay WANs.

The following are several situations in which tunneling (encapsulating traffic in another protocol) is useful:

- To enable multiprotocol local networks over a single-protocol backbone
- To provide workarounds for networks that use protocols that have limited hop counts; for example, RIP version 1, AppleTalk
- To connect discontiguous subnetworks
- To allow virtual private networks across WANs

Loopback Interfaces

You can specify a software-only interface called a loopback interface to emulate a physical interface. Loopback interfaces are supported on all platforms. A loopback interface is a virtual interface on a Cisco router that remains up (active) after you issue the **no shutdown** command until you disable it with the **shutdown** command. Unlike subinterfaces, loopback interfaces are independent of the state of any physical interface.

The loopback interface can be considered stable because once you enable it, it will remain up until you shut it down. This makes loopback interfaces ideal for assigning Layer 3 addresses such as IP addresses when you want a single address as a reference that is independent of the status of any physical interfaces in the networking device. A good example of this is using the IP address of a loopback interface as the IP address for the domain name system (DNS) host address for the networking device. Before loopback interfaces were available, network administrators had to configure a DNS host entry for every interface on a router that had an IP address assigned to it because they could never be certain which interface IP address might be available at any given time for managing the router. In the sample interface configuration and DNS entries for Router A shown below, you can see that there is a DNS entry for each interface.

Router A Interface Configuration Before Loopback

```
GigabitEthernet0 10.10.10.1 255.255.255.0
GigabitEthernet1 10.10.11.1 255.255.255.0
GigabitEthernet2 10.10.12.1 255.255.255.0
GigabitEthernet3 10.10.13.1 255.255.255.0
GigabitEthernet4 10.10.14.1 255.255.255.0
GigabitEthernet5 10.10.15.1 255.255.255.0
```

Router A DNS Entries Before Loopback

```
RouterA   IN  A  10.10.10.1
          IN  A  10.10.11.1
          IN  A  10.10.12.1
          IN  A  10.10.13.1
          IN  A  10.10.14.1
          IN  A  10.10.15.1
```

Interfaces on networking devices can fail, and they can also be taken out of service for maintenance. If any of the interfaces in Router A fails or is taken out of service, another networking device will not be able to access that interface. When you configure a networking device with a loopback interface and assign it an IP address that is advertised throughout the network, the networking device will be reachable by using this IP address as long as the networking device has at least one network interface capable of sending and receiving IP traffic. In the sample interface configuration and DNS entries for Router A after a loopback interface is configured, you can see that there is now only one DNS entry that can be used to reach the router over any of its physical interfaces.

Router A Interface Configuration After Loopback

```
Loopback 172.16.78.1 255.255.255.0
GigabitEthernet0 10.10.10.1 255.255.255.0
GigabitEthernet1 10.10.11.1 255.255.255.0
GigabitEthernet2 10.10.12.1 255.255.255.0
GigabitEthernet3 10.10.13.1 255.255.255.0
GigabitEthernet4 10.10.14.1 255.255.255.0
GigabitEthernet5 10.10.15.1 255.255.255.0
```

Router A DNS Entries After Loopback

```
RouterA   IN  A  172.16.78.1
```

The configured IP address of the loopback interface--172.16.78.1--can be used as the source address for packets generated by the router and forwarded to networking management applications and routing protocols. Unless this loopback interface is explicitly shut down, it is always reachable.

You can use the loopback interface as the termination address for open shortest path first (OSPF) or border gateway protocol (BGP) sessions. A loopback interface can also be used to establish a Telnet session from the console port of the device to its auxiliary port when all other interfaces are down. In applications where other routers or access servers attempt to reach this loopback interface, you should configure a routing protocol to distribute the subnet assigned to the loopback address.

IP Packets routed to the loopback interface are rerouted back to the router or access server and processed locally. IP packets routed out the loopback interface but not destined to the loopback interface are dropped. Under these two conditions, the loopback interface can behave like a null interface.

Loopback Interfaces Versus Loopback Mode

Loopback interfaces provide a stable source interface to ensure that the IP address assigned to the interface is always reachable as long as the IP routing protocols continue to advertise the subnet assigned to the loopback interface. Loopback mode, however, is used to test and diagnose issues with WAN (serial) links such as bit loss or data corruption. The idea is to configure a loop to return the data packets that were received by the interface back out the same interface to the device that originated the traffic. Loopback mode is used to troubleshoot problems by checking that the data packets are returned in the same condition in which they were sent. Errors in the data packets indicate a problem with the WAN infrastructure. Many types of serial interfaces have their own form of loopback command syntax that is entered under interface or controller configuration mode.

Null Interfaces

The null interface is a virtual network interface that is similar to the loopback interface. Whereas traffic to the loopback interface is directed to the router itself, traffic sent to the null interface is discarded. This interface is always up and can never forward or receive traffic; encapsulation always fails. The null interface functions similarly to the null devices available on most operating systems.

Null interfaces are used as a low-overhead method of discarding unnecessary network traffic. For example, if you do not want your network users to be able to reach certain IP subnets, you can create static IP routes for the subnets that point to the null interface of a networking device. Using the static IP routes takes less CPU time for the networking device than using IP access lists. The static-route configuration is also easier to configure than IP access lists because it is done in global configuration mode instead of in interface configuration mode.

The null interface may not be configured with an address. Traffic can be sent to this interface only by configuring a static route where the next hop is the null interface--represented by Null 0. One example of configuring the next hop to be the null interface is to create a route to an aggregate network that can then be announced through the BGP, or to ensure that traffic to a particular range of addresses is not propagated through the router, perhaps for security purposes.

The router always has a single null interface. By default, a packet sent to the null interface causes the router to respond by sending an Internet Control Message Protocol (ICMP) unreachable message to the source IP

address of the packet. You can configure the router either to send these responses or to drop the packets silently.

Subinterfaces

Subinterfaces are associated with physical interfaces. Subinterfaces are enabled when the physical interface with which they are associated is enabled, and subinterfaces are disabled when the physical interface is shut down.

Note Subinterfaces can be enabled and shut down independently of the physical port with which they are associated. However, you cannot enable a subinterface of a physical interface that has been shut down.

Subinterfaces are created by subdividing the physical interface into two or more virtual interfaces on which you can assign unique Layer 3 network addresses such as IP subnets. One of the first uses of subinterfaces was to resolve the problem with split horizon on Frame Relay WANs. Split horizon is a behavior associated with IP routing protocols such as RIP in which IP subnets are not advertised back out the same physical interface that they were learned over. Split horizon was implemented to prevent routing loops in IP networks. A routing loop can be created when the networking devices at both ends of a network connection advertise the same IP routes to each other. Split horizon was an issue for Frame Relay multipoint network interfaces--interfaces that connect to two or more remote networking devices over a single physical interface--because the default behavior of many networking devices was to implement split horizon, which means that the networking device did not advertise the IP routes that were learned over an interface back out the interface to other devices that were also reachable via the same physical interface. Subinterfaces were invented as a method of virtually subdividing a physical interface into two or more interfaces so that the IP routing protocols would see the network connection to each remote networking device as a separate physical interface even though the subinterfaces share a common physical interface. Although TCP/IP now disables split horizon limitations by default, protocols such as AppleTalk and IPX are still constrained by split horizon.

Subinterfaces are identified by a prefix that consists of the hardware interface descriptor (IDB) followed by a period and then by a number that is unique for that prefix. The full subinterface number must be unique to the networking device. For example, the first subinterface for GigabitEthernet interface 0/0/0 might be named GigabitEthernet 0/0/0.1 where .1 indicates the subinterface.

Tunnel Interfaces

Tunneling provides a way to encapsulate arbitrary packets inside a transport protocol. Tunnels are implemented as a virtual interface to provide a simple interface for configuration. The tunnel interface is not tied to specific "passenger" or "transport" protocols, but, rather, it is an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme.

There are several ways to implement tunnel interfaces depending on the connectivity that you need to provide. One common use for tunnels is to carry data traffic for a network protocol such as IPX over devices in your network that do not support IPX. For instance, if your network uses IPX in sites at the edge of your network but not in the core of your network, you can connect the IPX sites at the network edges by tunneling IPX in IP over the core of the network.

For more details about the various types of tunneling techniques available using Cisco IOS XE software, see the " Implementing Tunnels " module of the *Cisco IOS XE Interface and Hardware Component Configuration Guide.*

How to Configure Virtual Interfaces

Configuring a Loopback Interface

This task explains how to configure a loopback interface. A loopback interface can be considered stable because once you enable it, it will remain up until you shut it down. This makes loopback interfaces ideal for assigning Layer 3 addresses such as IP addresses when you want to have a single address to use as a reference that is independent of the status of any of the physical interfaces in the networking device.

Before you begin

The IP address for the loopback interface must be unique and not in use by another interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces loopback** *number*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *number* **Example:** `Router(config)# interface loopback 0`	Specifies a loopback interface and enters interface configuration mode. • Use the *number* argument to specify the number of the loopback interface that you want to create or configure. **Note** There is no limit on the number of loopback interfaces that you can create.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:**	Specifies an IP address for the loopback interface and enables IP processing on the interface.

	Command or Action	Purpose
	`Router(config-if)# ip address 10.20.1.2 255.255.255.0`	• Use the *ip-address* and *mask* arguments to specify the subnet for the loopback address.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show interfaces loopback** *number* **Example:** `Router# show interfaces loopback 0`	(Optional) Displays information about loopback interfaces. • Use the *number* argument to display information about one particular loopback interface. **Note** Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS Interface and Hardware Component Command Reference.
Step 7	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Examples

The following is sample output for the **show interfaces loopback** command.

```
Router# show interfaces loopback
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 10.20.1.2/24
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Configuring a Null Interface

This task explains how to configure a null interface. Null interfaces provide an alternative method to access control lists for filtering traffic. All unwanted traffic can be directed to the null interface; the null interface cannot receive or forward traffic, or allow its traffic to be encapsulated.

The only interface configuration command that you can specify for the null interface is the **no ip unreachables** command.

ICMP Unreachable Messages from Null Interfaces

To disable the sending of ICMP unreachable messages in response to packets sent to the null interface, use the **no ip unreachables** command in interface configuration mode. To reenable the sending of ICMP unreachable messages in response to packets sent to the null interface, use the **ip unreachables** command in interface configuration mode.

You can configure only one null interface on a device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface null** *number*
4. **no ip unreachables**
5. **end**
6. **show interfaces null** [*number*] [**accounting**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface null** *number* **Example:** `Device(config)# interface null 0`	Specifies a null interface and number, and enters interface configuration mode. • The number argument is always 0.
Step 4	**no ip unreachables** **Example:** `Device(config-if)# no ip unreachables`	Prevents the generation of ICMP unreachable messages on an interface. • This command affects all types of ICMP unreachable messages.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Exits to privileged EXEC mode.
Step 6	**show interfaces null** [*number*] [**accounting**] **Example:** `Device# show interfaces null 0`	(Optional) Displays information about null interfaces. • For null interfaces, the *number* argument is always 0. **Note** Only the syntax applicable to this task is used in this example. For more details, see the Cisco IOS Interface and Hardware Component Command Reference.

Examples

The following is sample output for the **show interfaces null** command.

```
Device# show interfaces null

Null0 is up, line protocol is up
  Hardware is Unknown
  MTU 1500 bytes, BW 10000000 Kbit, DLY 0 usec,
     reliability 0/255, txload 0/255, rxload 0/255
  Encapsulation ARPA, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Configuring a Subinterface

This task explains how to configure a subinterface. Subinterfaces can be enabled and shut down independently of the physical port with which they are associated. However, you cannot enable a subinterface of a physical interface that has been shut down.

Before you begin

The IP address for the interface must be unique and not in use by another interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number.subinterface-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces** *type number.subinterface-number*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number.subinterface-number* **Example:** `Router(config)# interface GigabitEthernet 2/3.5`	Specifies the interface type, interface number, and subinterface number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Specifies an IP address for the interface and enables IP processing on the interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show interfaces** *type number.subinterface-number* **Example:** `Router# show interfaces GigabitEthernet 2/3.5`	(Optional) Displays information about the interfaces.
Step 7	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Examples

The following is sample output from the **show interfaces**command:

```
Router# show interfaces GigabitEthernet 2/3.5
GigabitEthernet2/3.5432 is down, line protocol is down (notconnect)
  Hardware is c7600 1Gb 802.3, address is 001b.0de6.c100 (bia 001b.0de6.c100)
  Description: *sample*
  Internet address is 10.11.12.13/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  2339.
  ARP type: ARPA, ARP Timeout 04:00:00
  Keepalive set (10 sec)
  Last clearing of "show interface" counters never
```

Configuring a Subinterface

This task explains how to configure a subinterface. Subinterfaces can be enabled and shut down independently of the physical port with which they are associated. However, you cannot enable a subinterface of a physical interface that has been shut down.

Before you begin

The IP address for the interface must be unique and not in use by another interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number.subinterface-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces** *type number.subinterface-number*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number.subinterface-number* **Example:** `Router(config)# interface GigabitEthernet 2/3.5`	Specifies the interface type, interface number, and subinterface number and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	Specifies an IP address for the interface and enables IP processing on the interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 6	**show interfaces** *type number.subinterface-number* **Example:** `Router# show interfaces GigabitEthernet 2/3.5`	(Optional) Displays information about the interfaces.
Step 7	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Examples

The following is sample output from the **show interfaces**command:

```
Router# show interfaces GigabitEthernet 2/3.5
GigabitEthernet2/3.5432 is down, line protocol is down (notconnect)
  Hardware is c7600 1Gb 802.3, address is 001b.0de6.c100 (bia 001b.0de6.c100)
  Description: *sample*
  Internet address is 10.11.12.13/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  2339.
  ARP type: ARPA, ARP Timeout 04:00:00
  Keepalive set (10 sec)
  Last clearing of "show interface" counters never
```

Configuring Logical Layer 3 VLAN Interfaces

Before you begin

Before you configure logical Layer 3 VLAN interfaces, you must create and configure the VLANs on the device, assign VLAN membership to the Layer 2 interfaces, enable IP routing if IP routing is disabled, and specify an IP routing protocol.

SUMMARY STEPS

1. **configure terminal**
2. **interface vlan** *vlan ID*
3. **ip address** *ip_address subnet_mask*
4. **no shutdown**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface vlan** *vlan ID* **Example:** `Device(config)# interface vlan vlan_ID`	Selects an interface to configure.
Step 3	**ip address** *ip_address subnet_mask* **Example:** `Device(config-if)# ip address ip_address subnet_mask`	Configures the IP address and IP subnet.
Step 4	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits the configuration mode.

Examples

The following is sample output from the **show interface vlan**command, which displays the interface IP address configuration and status of Layer 3 VLAN interface vlan 2

```
Device# show interface vlan
 Vlan2 is up, line protocol is down
   Hardware is Ethernet SVI, address is 00D.588F.B604 (bia 00D.588F.B604)
```

```
   Internet address is 172.20.52.106/29
   MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
      reliability 255/255, txload 1/255, rxload 1/255
   Encapsulation ARPA, loopback not set
   ARP type: ARPA, ARP Timeout 04:00:00
   Last input never, output never, output hang never
   Last clearing of "show interface" counters never
   Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
   Queueing strategy: fifo
   Output queue: 0/40 (size/max)
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
      0 packets input, 0 bytes, 0 no buffer
      Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
      0 packets output, 0 bytes, 0 underruns
      0 output errors, 0 interface resets
      0 output buffer failures, 0 output buffers swapped out
```

Configuration Examples for Virtual Interfaces

Example Configuring a Loopback Interface

The following example shows the configuration sequence of a loopback interface, loopback 0:

```
interface loopback 0
 ip address 209.165.200.225 255.255.255.0
 end
```

Example Configuring a Null Interface

The following example shows the configuration sequence of a null interface and how to drop the ICMP unreachable messages. All packets sent to the null interface are dropped and in this example, the ICMP messages usually sent in response to packets being sent to the null interface are dropped.

```
interface null 0
 no ip unreachables
 end
```

Example Configuring a Subinterface

The following example shows the configuration sequence of a subinterface:

```
interface GigabitEthernet 2/3.5
 description *sample*
 encapsulation dot1Q 2339
 ip address 209.165.200.225 255.255.255.224
end
```

Where to Go Next

- If you want to implement tunnels in your network, see the " Implementing Tunnels " module of the *Cisco IOS XE Interface and Hardware Component Configuration Guide* .
- If you want to implement physical (hardware) interfaces (such as Gigabit Ethernet or serial interfaces) in your network, see the " Configuring Physical Interfaces " module of the *Cisco IOS XE Interface and Hardware Component Configuration Guide* .

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Interface commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS Interface and Hardware Component Command Reference*
Cisco IOS XE Interface and Hardware Component configuration modules	*Cisco IOS XE Interface and Hardware Component Configuration Guide*
Configuration example showing how to use loopback interfaces with BGP	Sample Configuration for iBGP and eBGP With or Without a Loopback Address

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 256

Implementing Tunnels

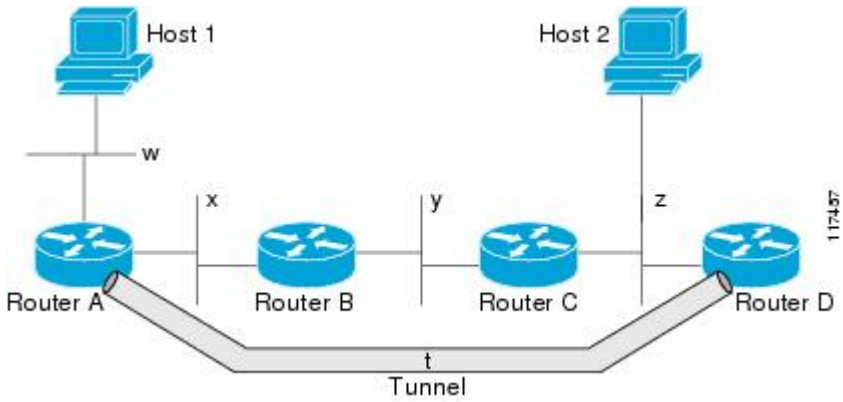
This module describes the various types of tunneling techniques. Configuration details and examples are provided for the tunnel types that use physical or virtual interfaces. Many tunneling techniques are implemented using technology-specific commands, and links are provided to the appropriate technology modules.

Tunneling provides a way to encapsulate arbitrary packets inside a transport protocol. Tunnels are implemented as virtual interfaces to provide a simple interface for configuration purposes. The tunnel interface is not tied to specific "passenger" or "transport" protocols, but rather is an architecture to provide the services necessary to implement any standard point-to-point encapsulation scheme.

Restrictions for Implementing Tunnels

- It is important to allow the tunnel protocol to pass through a firewall and access control list (ACL) check.
- Multiple point-to-point tunnels can saturate the physical link with routing information if the bandwidth is not configured correctly on a tunnel interface.
- A tunnel looks like a single hop link, and routing protocols may prefer a tunnel over a multihop physical path. The tunnel, despite looking like a single hop link, may traverse a slower path than a multihop link. A tunnel is as robust and fast, or as unreliable and slow, as the links that it actually traverses. Routing protocols that make their decisions based only on hop counts will often prefer a tunnel over a set of physical links. A tunnel might appear to be a one-hop, point-to-point link and have the lowest-cost path, but the tunnel may actually cost more in terms of latency when compared to an alternative physical topology. For example, in the topology shown in the figure below, packets from Host 1 will appear to travel across networks w, t, and z to get to Host 2 instead of taking the path w, x, y, and z because the tunnel hop count appears shorter. In fact, the packets going through the tunnel will still be traveling across Router A, B, and C, but they must also travel to Router D before coming back to Router C.

Figure 214: Tunnel Precautions: Hop Counts



- A tunnel may have a recursive routing problem if routing is not configured accurately. The best path to a tunnel destination is via the tunnel itself; therefore recursive routing causes the tunnel interface to flap. To avoid recursive routing problems, keep the control-plane routing separate from the tunnel routing by using the following methods:
  - Use a different autonomous system number or tag.
  - Use a different routing protocol.
  - Ensure that static routes are used to override the first hop (watch for routing loops).

  The following error is displayed when there is recursive routing to a tunnel destination:

```
%TUN-RECURDOWN Interface Tunnel 0
temporarily disabled due to recursive routing
```

Information About Implementing Tunnels

Tunneling Versus Encapsulation

To understand how tunnels work, you must be able to distinguish between concepts of encapsulation and tunneling. Encapsulation is the process of adding headers to data at each layer of a particular protocol stack. The Open Systems Interconnection (OSI) reference model describes the functions of a network. To send a data packet from one host (for example, a PC) to another on a network, encapsulation is used to add a header in front of the data packet at each layer of the protocol stack in descending order. The header must contain a data field that indicates the type of data encapsulated at the layer immediately above the current layer. As the packet ascends the protocol stack on the receiving side of the network, each encapsulation header is removed in reverse order.

Tunneling encapsulates data packets from one protocol within a different protocol and transports the packets on a foreign network. Unlike encapsulation, tunneling allows a lower-layer protocol and a same-layer protocol to be carried through the tunnel. A tunnel interface is a virtual (or logical) interface. Tunneling consists of three main components:

- Passenger protocol—The protocol that you are encapsulating. For example, IPv4 and IPv6 protocols.
- Carrier protocol—The protocol that encapsulates. For example, generic routing encapsulation (GRE) and Multiprotocol Label Switching (MPLS).

- Transport protocol--The protocol that carries the encapsulated protocol. The main transport protocol is IP.

Tunnel ToS

Tunnel type of service (ToS) allows you to tunnel network traffic and group all packets in the same ToS byte value. The ToS byte values and Time-to-Live (TTL) hop-count value can be set in the encapsulating IP header of tunnel packets for an IP tunnel interface on a router. Tunnel ToS feature is supported for Cisco Express Forwarding (formerly known as CEF), fast switching, and process switching.

The ToS and TTL byte values are defined in RFC 791. RFC 2474, and RFC 2780 obsolete the use of the ToS byte as defined in RFC 791. RFC 791 specifies that bits 6 and 7 of the ToS byte (the first two least significant bits) are reserved for future use and should be set to 0.

EoMPLS over GRE

Ethernet over MPLS (EoMPLS) is a tunneling mechanism that allows you to tunnel Layer 2 traffic through a Layer 3 MPLS network. EoMPLS is also known as Layer 2 tunneling.

EoMPLS effectively facilitates Layer 2 extension over long distances. EoMPLS over GRE helps you to create the GRE tunnel as hardware-based switched, and encapsulates EoMPLS frames within the GRE tunnel. The GRE connection is established between the two core routers, and then the MPLS label switched path (LSP) is tunneled over.

GRE encapsulation is used to define a packet that has header information added to it prior to being forwarded. De-encapsulation is the process of removing the additional header information when the packet reaches the destination tunnel endpoint.

When a packet is forwarded through a GRE tunnel, two new headers are added to the front of the packet and hence the context of the new payload changes. After encapsulation, what was originally the data payload and separate IP header are now known as the GRE payload. A GRE header is added to the packet to provide information on the protocol type and the recalculated checksum. A new IP header is also added to the front of the GRE header. This IP header contains the destination IP address of the tunnel.

The GRE header is added to packets such as IP, Layer 2 VPN, and Layer 3 VPN before the header enters into the tunnel. All routers along the path that receives the encapsulated packet use the new IP header to determine how the packet can reach the tunnel endpoint.

In IP forwarding, on reaching the tunnel destination endpoint, the new IP header and the GRE header are removed from the packet and the original IP header is used to forward the packet to the final destination.

The EoMPLS over GRE feature removes the new IP header and GRE header from the packet at the tunnel destination, and the MPLS label is used to forward the packet to the appropriate Layer 2 attachment circuit or Layer 3 VRF.

The scenarios in the following sections describe the L2VPN and L3VPN over GRE deployment on provider edge (PE) or provider (P) routers:

Provider Edge to Provider Edge Generic Routing EncapsulationTunnels

In the Provider Edge to Provider Edge (PE) GRE tunnels scenario, a customer does not transition any part of the core to MPLS but prefers to offer EoMPLS and basic MPLS VPN services. Therefore, GRE tunneling of MPLS traffic is done between PEs.

Provider to Provider Generic Routing Encapsulation Tunnels

In the Provider to Provider (P) GRE tunnels scenario, Multiprotocol Label Switching (MPLS) is enabled between Provider Edge (PE) and P routers but the network core can either have non-MPLS aware routers or IP encryption boxes. In this scenario, GRE tunneling of the MPLS labeled packets is done between P routers.

Provider Edge to Provider Generic Routing Encapsulation Tunnels

In a Provider Edge to Provider GRE tunnels scenario, a network has MPLS-aware P to P nodes. GRE tunneling is done between a PE to P non-MPLS network segment.

Features Specific to Generic Routing Encapsulation

You should understand the following configurations and information for a deployment scenario:

- Tunnel endpoints can be loopbacks or physical interfaces.
- Configurable tunnel keepalive timer parameters per endpoint and a syslog message must be generated when the keepalive timer expires.
- Bidirectional forwarding detection (BFD) is supported for tunnel failures and for the Interior Gateway Protocol (IGP) that use tunnels.
- IGP load sharing across a GRE tunnel is supported.
- IGP redundancy across a GRE tunnel is supported.
- Fragmentation across a GRE tunnel is supported.
- Ability to pass jumbo frames is supported.
- All IGP control plane traffic is supported.
- IP ToS preservation across tunnels is supported.
- A tunnel should be independent of the endpoint physical interface type; for example, ATM, Gigabit, Packet over SONET (POS), and TenGigabit.
- Up to 100 GRE tunnels are supported.

Features Specific to Ethernet over MPLS

- Any Transport over MPLS (AToM) sequencing.
- IGP load sharing and redundancy.
- Port mode Ethernet over MPLS (EoMPLS).
- Pseudowire redundancy.
- Support for up to to 200 EoMPLS virtual circuits (VCs).
- Tunnel selection and the ability to map a specific pseudowire to a GRE tunnel.
- VLAN mode EoMPLS.

Features Specific to Multiprotocol Label Switching Virtual Private Network

- Support for the PE role with IPv4 VRF.
- Support for all PE to customer edge (CE) protocols.
- Load sharing through multiple tunnels and also equal cost IGP paths with a single tunnel.
- Support for redundancy through unequal cost IGP paths with a single tunnel.
- Support for the IP precedence value being copied onto the expression (EXP) bits field of the Multiprotocol Label Switching (MPLS) label and then onto the precedence bits on the outer IPv4 ToS field of the generic routing encapsulation (GRE) packet.

See the section, Example: Configuring EoMPLS over GRE, on page 3084" for a sample configuration sequence of EoMPLS over GRE. For more details on EoMPLS over GRE, see the Deploying and Configuring MPLS Virtual Private Networks In IP Tunnel Environments document.

Path MTU Discovery

Path MTU Discovery (PMTUD) can be enabled on a GRE or IP-in-IP tunnel interface. When PMTUD (RFC 1191) is enabled on a tunnel interface, the router performs PMTUD processing for the GRE (or IP-in-IP) tunnel IP packets. The router always performs PMTUD processing on the original data IP packets that enter the tunnel. When PMTUD is enabled, packet fragmentation is not permitted for packets that traverse the tunnel because the Don't Fragment (DF) bit is set on all the packets. If a packet that enters the tunnel encounters a link with a smaller MTU, the packet is dropped and an Internet Control Message Protocol (ICMP) message is sent back to the sender of the packet. This message indicates that fragmentation was required (but not permitted) and provides the MTU of the link that caused the packet to be dropped.

Note PMTUD on a tunnel interface requires that the tunnel endpoint be able to receive ICMP messages generated by routers in the path of the tunnel. Ensure that ICMP messages can be received before using PMTUD over firewall connections.

Use the **tunnel path-mtu-discovery** command to enable PMTUD for the tunnel packets and use the **show interfaces tunnel** command to verify the tunnel PMTUD parameters. PMTUD works only on GRE and IP-in-IP tunnel interfaces.

QoS Options for Tunnels

A tunnel interface supports various quality of service (QoS) features as a physical interface. QoS provides a way to ensure that mission-critical traffic has an acceptable level of performance. QoS options for tunnels include support for applying generic traffic shaping (GTS) directly on the tunnel interface and support for class-based shaping using the modular QoS CLI (MQC). Tunnel interfaces also support class-based policing, but they do not support committed access rate (CAR).

GRE tunnels allow the router to copy the IP precedence bit values of the ToS byte to the tunnel or the GRE IP header that encapsulates the inner packet. Intermediate routers between the tunnel endpoints can use the IP precedence values to classify packets for QoS features such as policy routing, weighted fair queueing (WFQ), and weighted random early detection (WRED).

When packets are encapsulated by tunnel or encryption headers, QoS features are unable to examine the original packet headers and correctly classify the packets. Packets that travel across the same tunnel have the same tunnel headers, so the packets are treated identically if the physical interface is congested. Tunnel packets can, however, be classified before tunneling and encryption can occur when a user applies the QoS preclassify feature on the tunnel interface or on the crypto map.

Note Class-based WFQ (CBWFQ) inside class-based shaping is not supported on a multipoint interface.

For examples of how to implement some QoS features on a tunnel interface, see the sectionConfiguring QoS Options on Tunnel Interfaces Examples, on page 3086.

How to Implement Tunnels

Determining the Tunnel Type

Before configuring a tunnel, you must determine the type of tunnel you want to create.

SUMMARY STEPS

1. Determine the passenger protocol. A passenger protocol is the protocol that you are encapsulating.
2. Determine the **tunnel mode** command keyword, if appropriate.

DETAILED STEPS

Step 1 Determine the passenger protocol. A passenger protocol is the protocol that you are encapsulating.

Step 2 Determine the **tunnel mode** command keyword, if appropriate.

The table below shows how to determine the appropriate keyword to be used with the **tunnel mode** command.

Table 310: Determining the tunnel mode Command Keyword

Keyword	Purpose
dvmrp	Use the **dvmrp** keyword to specify that the Distance Vector Multicast Routing Protocol encapsulation will be used.
gre ip	Use the **gre** and **ip** keywords to specify that GRE encapsulation over IP will be used.
gre ipv6	Use the **gre** and **ipv6** keywords to specify that GRE encapsulation over IPv6 will be used.
ipip [**decapsulate-any**]	Use the **ipip** keyword to specify that IP-in-IP encapsulation will be used. The optional **decapsulate-any** keyword terminates any number of IP-in-IP tunnels at one tunnel interface. Note that this tunnel will not carry any outbound traffic; however, any number of remote tunnel endpoints can use a tunnel configured as their destination.
ipv6	Use the **ipv6** keyword to specify that generic packet tunneling in IPv6 will be used.

Keyword	Purpose
ipv6ip	Use the **ipv6ip** keyword to specify that IPv6 will be used as the passenger protocol and IPv4 as both the carrier (encapsulation) and transport protocol. When additional keywords are not used, manual IPv6 tunnels are configured. Additional keywords can be used to specify IPv4-compatible, 6to4, or ISATAP tunnels.
mpls	Use the **mpls** keyword to specify that MPLS will be used for configuring traffic engineering (TE) tunnels.

Configuring an IPv4 GRE Tunnel

Perform this task to configure a GRE tunnel. A tunnel interface is used to pass protocol traffic across a network that does not normally support the protocol. To build a tunnel, you must define a tunnel interface on each of the two routers, and the tunnel interfaces must reference each other. At each router, the tunnel interface must be configured with a Layer 3 address. The tunnel endpoints, tunnel source, and tunnel destination must be defined, and the type of tunnel must be selected. Optional steps can be performed to customize the tunnel.

Remember to configure the router at each end of the tunnel. If only one side of a tunnel is configured, the tunnel interface may still come up and stay up (unless keepalive is configured), but packets going into the tunnel will be dropped.

GRE Tunnel Keepalive

Keepalive packets can be configured to be sent over IP-encapsulated GRE tunnels. You can specify the rate at which keepalives are sent and the number of times that a device will continue to send keepalive packets without a response before the interface becomes inactive. GRE keepalive packets may be sent from both sides of a tunnel or from just one side.

Before you begin

Ensure that the physical interface to be used as the tunnel source in this task is up and configured with the appropriate IP address. For hardware technical descriptions and information about installing interfaces, see the hardware installation and configuration publication for your product.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bandwidth** *kb/s*
5. **keepalive** [*period* [*retries*]]
6. **tunnel source** {*ip-address* | *interface-type interface-number*}
7. **tunnel destination** {*hostname* | *ip-address*}
8. **tunnel key** *key-number*
9. **tunnel mode gre** { **ip** | **multipoint**}
10. **ip mtu** *bytes*
11. **tunnel mpls-ip-only**

12. **ip tcp mss** *mss-value*
13. **tunnel path-mtu-discovery** [**age-timer** {*aging-mins* | **infinite**}]
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 0`	Specifies the interface type and number, and enters interface configuration mode. • To configure a tunnel, use **tunnel** for the *type* argument.
Step 4	**bandwidth** *kb/s* **Example:** `Router(config-if)# bandwidth 1000`	Sets the current bandwidth value for an interface and communicates it to higher-level protocols. • Specifies the tunnel bandwidth to be used to transmit packets. • Use the *kb/s* argument to set the bandwidth, in kilobits per second (kb/s). **Note** This is only a routing parameter; it does not affect the physical interface. The default bandwidth setting on a tunnel interface is 9.6 kb/s. You should set the bandwidth on a tunnel to an appropriate value.
Step 5	**keepalive** [*period* [*retries*]] **Example:** `Router(config-if)# keepalive 3 7`	(Optional) Specifies the number of times the device will continue to send keepalive packets without response before bringing the tunnel interface protocol down. • GRE keepalive packets may be configured either on only one side of the tunnel or on both. • If GRE keepalive is configured on both sides of the tunnel, the *period* and *retries* arguments can be different at each side of the link. **Note** This command is supported only on GRE point-to-point tunnels.

	Command or Action	Purpose
Step 6	**tunnel source** {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Configures the tunnel source. • Use the *ip-address* argument to specify the source IP address. • Use the *interface-type* and *interface-number* arguments to specify the interface to be used. **Note** The tunnel source IP address and destination IP addresses must be defined on two separate devices.
Step 7	**tunnel destination** {*hostname* | *ip-address*} **Example:** `Router(config-if)# tunnel destination 10.0.2.1`	Configures the tunnel destination. • Use the *hostname* argument to specify the name of the host destination. • Use the *ip-address* argument to specify the IP address of the host destination. **Note** The tunnel source and destination IP addresses must be defined on two separate devices.
Step 8	**tunnel key** *key-number* **Example:** `Router(config-if)# tunnel key 1000`	(Optional) Enables an ID key for a tunnel interface. • Use the *key-number* argument to identify a tunnel key that is carried in each packet. • Tunnel ID keys can be used as a form of weak security to prevent improper configuration or injection of packets from a foreign source. **Note** This command is supported only on GRE tunnel interfaces. We do not recommend relying on this key for security purposes.
Step 9	**tunnel mode gre** { **ip** | **multipoint**} **Example:** `Device(config-if)# tunnel mode gre ip`	Specifies the encapsulation protocol to be used in the tunnel. • Use the **gre ip** keywords to specify that GRE over IP encapsulation will be used. • Use the **gre multipoint** keywords to specify that multipoint GRE (mGRE) will be used.
Step 10	**ip mtu** *bytes* **Example:** `Device(config-if)# ip mtu 1400`	(Optional) Sets the MTU size of IP packets sent on an interface. • If an IP packet exceeds the MTU set for the interface, the Cisco software will fragment it unless the DF bit is set. • All devices on a physical medium must have the same protocol MTU in order to operate.

	Command or Action	Purpose
		• For IPv6 packets, use the **ipv6 mtu** command. **Note** If the **tunnel path-mtu-discovery** command is enabled do not configure this command.
Step 11	**tunnel mpls-ip-only** **Example:** `Device(config-if)# tunnel mpls-ip-only`	Copies the Do Not Fragment bit from the inner IP header to the IP header of the tunnel packet. If the Do Not Fragment bit is not set, and the IP packet exceeds the MTU set for the interface, the payload is fragmented. When **tunnel path-mtu-discovery** is enabled, the **tunnel path-mtu-discovery** automatically gets enabled due to the dependency.
Step 12	**ip tcp mss** *mss-value* **Example:** `Device(config-if)# ip tcp mss 250`	(Optional) Specifies the maximum segment size (MSS) for TCP connections that originate or terminate on a router. • Use the *mss-value* argument to specify the maximum segment size for TCP connections, in bytes.
Step 13	**tunnel path-mtu-discovery** [**age-timer** {*aging-mins* | **infinite**}] **Example:** `Device(config-if)# tunnel path-mtu-discovery`	(Optional) Enables PMTUD on a GRE or IP-in-IP tunnel interface. • When PMTUD is enabled on a tunnel interface, PMTUD will operate for GRE IP tunnel packets to minimize fragmentation in the path between the tunnel endpoints.
Step 14	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

What to Do Next

Proceed to the “Verifying Tunnel Configuration and Operation” section.

Configuring 6to4 Tunnels

Before you begin

With 6to4 tunnels, the tunnel destination is determined by the border-router IPv4 address, which is concatenated to the prefix 2002::/16 in the format 2002:*border-router-IPv4-address* ::/48. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks.

Note The configuration of only one IPv4-compatible tunnel and one 6to4 IPv6 tunnel is supported on a router. If you choose to configure both of these tunnel types on the same router, Cisco recommends that they not share the same tunnel source.

A 6to4 tunnel and an IPv4-compatible tunnel cannot share the same interface because both of them are NBMA "point-to-multipoint" access links, and only the tunnel source can be used to reorder the packets from a multiplexed packet stream into a single packet stream for an incoming interface. When a packet with an IPv4 protocol type of 41 arrives on an interface, the packet is mapped to an IPv6 tunnel interface on the basis of the IPv4 address. However, if both the 6to4 tunnel and the IPv4-compatible tunnel share the same source interface, the router cannot determine the IPv6 tunnel interface to which it should assign the incoming packet.

Manually configured IPv6 tunnels can share the same source interface because a manual tunnel is a "point-to-point" link, and both IPv4 source and the IPv4 destination of the tunnel are defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *interface-type interface-number*}
6. **tunnel mode ipv6ip 6to4**
7. **exit**
8. **ipv6 route** *ipv6-prefix* / *prefix-length* **tunnel** *tunnel-number*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] **Example:**	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.

	Command or Action	Purpose
	`Router(config-if)# ipv6 address 2002:c0a8:6301:1::1/64`	• The 32 bits following the initial 2002::/16 prefix correspond to an IPv4 address assigned to the tunnel source. **Note** See the "Configuring Basic Connectivity for IPv6" module for more information on configuring IPv6 addresses.
Step 5	**tunnel source** {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. **Note** The interface type and number specified in the **tunnel source** command must be configured with an IPv4 address.
Step 6	**tunnel mode ipv6ip 6to4** **Example:** `Router(config-if)# tunnel mode ipv6ip 6to4`	Specifies an IPv6 overlay tunnel using a 6to4 address.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**ipv6 route** *ipv6-prefix* / *prefix-length* **tunnel** *tunnel-number* **Example:** `Router(config)# ipv6 route 2002::/16 tunnel 0`	Configures a static route to the specified tunnel interface. **Note** When configuring a 6to4 overlay tunnel, you must configure a static route for the IPv6 6to4 prefix 2002::/16 to the 6to4 tunnel interface. • The tunnel number specified in the **ipv6 route** command must be the same tunnel number specified in the **interface tunnel** command.
Step 9	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

What to Do Next

Proceed to the “Verifying Tunnel Configuration and Operation” section.

Verifying Tunnel Configuration and Operation

The **show** and **ping** commands in the steps below can be used in any sequence. The following commands can be used for GRE tunnels, IPv6 manually configured tunnels, and IPv6 over IPv4 GRE tunnels.

SUMMARY STEPS

1. **enable**
2. **show interfaces tunnel** *number* [**accounting**]
3. **ping** [*protocol*] *destination*
4. **show ip route** [*address* [*mask*]]
5. **ping** [*protocol*] *destination*

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 show interfaces tunnel *number* [accounting]

Two routers are configured to be endpoints of a tunnel. Device A has Gigabit Ethernet interface 0/0/0 configured as the source for tunnel interface 0 with an IPv4 address of 10.0.0.1 and an IPv6 prefix of 2001:0DB8:1111:2222::1/64. Device B has Gigabit Ethernet interface 0/0/0 configured as the source for tunnel interface 1 with an IPv4 address of 10.0.0.2 and an IPv6 prefix of 2001:0DB8:1111:2222::2/64.

To verify that the tunnel source and destination addresses are configured, use the **show interfaces tunnel** command on Device A.

Example:

```
Device A# show interfaces tunnel 0

Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.1 (GigabitEthernet0/0/0), destination 10.0.0.2, fastswitch TTL 255
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Tunnel TTL 255
  Checksumming of packets disabled,  fast tunneling enabled
  Last input 00:00:14, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4 packets input, 352 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     8 packets output, 704 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Step 3 ping [*protocol*] *destination*

To check that the local endpoint is configured and working, use the **ping** command on Device A.

Example:

```
DeviceA# ping 2001:0DB8:1111:2222::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:0DB8:1111:2222::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

Step 4 **show ip route** [*address* [*mask*]]

To check that a route exists to the remote endpoint address, use the **show ip route** command.

Example:

```
DeviceA# show ip route 10.0.0.2

Routing entry for 10.0.0.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via GigabitEthernet0/0/0
      Route metric is 0, traffic share count is 1
```

Step 5 **ping** [*protocol*] *destination*

To check that the remote endpoint address is reachable, use the **ping** command on Device A.

Note The remote endpoint address may not be reachable using the **ping** command because of filtering, but the tunnel traffic may still reach its destination.

Example:

```
DeviceA# ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/28 ms
```

To check that the remote IPv6 tunnel endpoint is reachable, use the **ping** command again on Device A. The note regarding filtering earlier in step also applies to this example.

Example:

```
DeviceA# ping 2001:0DB8:1111:2222::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

These steps may be repeated at the other endpoint of the tunnel.

Configuration Examples for Implementing Tunnels

Example: Configuring a GRE IPv4 Tunnel

The following example shows a simple configuration of GRE tunneling. Note that Gigabit Ethernet interface 0/0/1 is the tunnel source for Router A and the tunnel destination for Router B. Fast Ethernet interface 0/0/1 is the tunnel source for Router B and the tunnel destination for Router A.

Router A

```
interface Tunnel 0
 ip address 10.1.1.2 255.255.255.0
 tunnel source GigabitEthernet 0/0/1
 tunnel destination 192.168.3.2
 tunnel mode gre ip
!
interface GigabitEthernet 0/0/1
 ip address 192.168.4.2 255.255.255.0
```

Router B

```
interface Tunnel 0
 ip address 10.1.1.1 255.255.255.0
 tunnel source FastEthernet 0/0/1
 tunnel destination 192.168.4.2
 tunnel mode gre ip
!
interface FastEthernet 0/0/1
 ip address 192.168.3.2 255.255.255.0
```

The following example configures a GRE tunnel running both IS-IS and IPv6 traffic between Router A and Router B:

Router A

```
ipv6 unicast-routing
clns routing
!
interface Tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::1/64
 ipv6 router isis
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.0.0.2
 tunnel mode gre ip
!
interface GigabitEthernet 0/0/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 network 49.0000.0000.000a.00
```

Router B

```
ipv6 unicast-routing
clns routing
!
interface Tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::2/64
 ipv6 router isis
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.0.0.1
 tunnel mode gre ip
!
interface GigabitEthernet 0/0/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 network 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

Example: Configuring EoMPLS over GRE

Router A Configuration

```
vrf definition VPN1
 rd 100:1
 address-family ipv4
 route-target both 100:1
 exit-address-family
 !
mpls label protocol ldp
mpls ldp neighbor 209.165.200.224 targeted
mpls ldp router-id Loopback0 force
!
interface tunnel 0
 ip address 209.165.200.225 255.255.255.224
 mpls label protocol ldp
 mpls ip
 keepalive 10 3
 tunnel source TenGigabitEthernet 2/1/0
 tunnel destination 209.165.200.226
!
interface Loopback 0
 ip address 209.165.200.230 255.255.255.224
 !
interface TenGigabitEthernet 2/1/0
 mtu 9216
 ip address 209.165.200.235 255.255.255.224
!
interface TenGigabitEthernet 9/1
 no ip address
!
interface TenGigabitEthernet 9/1.11
 vrf forwarding VPN1
 encapsulation dot1Q 300
 ip address 209.165.200.237 255.255.255.224
!
interface TenGigabitEthernet 9/2
```

```
 mtu 9216
 no ip address
xconnect 209.165.200.239 200 encapsulation mpls
!
router bgp 65000
 bgp log-neighbor-changes
 neighbor 209.165.200.240 remote-as 65000
 neighbor 209.165.200.240 update-source Loopback0
neighbor 209.165.200.245 remote-as 100
 !
address-family vpnv4
  neighbor 209.165.200.240 activate
  neighbor 209.165.200.240 send-community extended
 !
 address-family ipv4 vrf VPN1
  no synchronization
  neighbor 209.165.200.247 remote-as 100
  neighbor 209.165.200.248 activate
  neighbor 209.165.200.249 send-community extended
!
ip route 209.165.200.251 255.255.255.224 tunnel 0
ip route 209.165.200.254 255.255.255.224 209.165.200.256
Router B Configuration
vrf definition VPN1
 rd 100:1
 address-family ipv4
 route-target both 100:1
exit-address-family
 !
mpls ldp neighbor 209.165.200.229 targeted
mpls label protocol ldp
mpls ldp router-id Loopback0 force
!
interface tunnel 0
 ip address 209.165.200.230 255.255.255.224
 mpls label protocol ldp
 mpls ip
 keepalive 10 3
 tunnel source TenGigabitEthernet 3/3/0
 tunnel destination 209.165.200.232
!
interface Loopback 0
 ip address 209.165.200.234 255.255.255.224
!
interface TenGigabitEthernet 2/1/1
 mtu 9216
 no ip address
 xconnect 209.165.200.237 200 encapsulation mpls
!
interface TenGigabitEthernet 2/3/1
 mtu 9216
 no ip address
!
interface TenGigabitEthernet 2/3.11/1
 vrf forwarding VPN1
 encapsulation dot1Q 300
 ip address 209.165.200.239 255.255.255.224
!
interface TenGigabitEthernet 3/3/0
 mtu 9216
 ip address 209.165.200.240 255.255.255.224
!
router bgp 65000
 bgp log-neighbor-changes
```

```
 neighbor 209.165.200.241 remote-as 65000
 neighbor 209.165.200.241 update-source Loopback0
 neighbor 209.165.200.244 remote-as 200
 !
 address-family vpnv4
  neighbor 209.165.200.241 activate
  neighbor 209.165.200.241 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPN1
  no synchronization
  neighbor 209.165.200.246 remote-as 200
  neighbor 209.165.200.246 activate
  neighbor 209.165.200.246 send-community extended
 exit-address-family
¡
ip route 209.165.200.226 255.255.255.224 tunnel 0
ip route 209.165.200.229 255.255.255.224 209.165.200.235
```

Configuring QoS Options on Tunnel Interfaces Examples

The following sample configuration applies GTS directly on the tunnel interface. In this example, the configuration shapes the tunnel interface to an overall output rate of 500 kb/s.

```
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
 traffic-shape rate 500000 125000 125000 1000
 tunnel source 10.1.1.1
 tunnel destination 10.2.2.2
```

The following sample configuration shows how to apply the same shaping policy to the tunnel interface with the MQC commands:

```
policy-map tunnel
 class class-default
 shape average 500000 125000 125000
!
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
 service-policy output tunnel
 tunnel source 10.1.35.1
 tunnel destination 10.1.35.2
```

Configuring QoS Options on Tunnel Interfaces Examples

The following sample configuration applies GTS directly on the tunnel interface. In this example, the configuration shapes the tunnel interface to an overall output rate of 500 kb/s.

```
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
 traffic-shape rate 500000 125000 125000 1000
 tunnel source 10.1.1.1
 tunnel destination 10.2.2.2
```

The following sample configuration shows how to apply the same shaping policy to the tunnel interface with the MQC commands:

```
policy-map tunnel
 class class-default
 shape average 500000 125000 125000
!
interface Tunnel 0
 ip address 10.1.2.1 255.255.255.0
 service-policy output tunnel
 tunnel source 10.1.35.1
 tunnel destination 10.1.35.2
```

Policing Example

When an interface becomes congested and packets start to queue, you can apply a queueing method to packets that are waiting to be transmitted. Logical interfaces--tunnel interfaces in this example--do not inherently support a state of congestion and do not support the direct application of a service policy that applies a queueing method. Instead, you must apply a hierarchical policy. Create a "child" or lower-level policy that configures a queueing mechanism, such as low-latency queueing, with the **priority** command and CBWFQ with the **bandwidth** command.

```
policy-map child
 class voice
 priority 512
```

Create a "parent" or top-level policy that applies class-based shaping. Apply the child policy as a command under the parent policy because admission control for the child class is done according to the shaping rate for the parent class.

```
policy-map tunnel
 class class-default
 shape average 2000000
 service-policy child
```

Apply the parent policy to the tunnel interface.

```
interface tunnel 0
 service-policy tunnel
```

In the following example, a tunnel interface is configured with a service policy that applies queueing without shaping. A log message is displayed noting that this configuration is not supported.

```
Router(config)# interface tunnel1
Router(config-if)# service-policy output child
Class Based Weighted Fair Queueing not supported on this interface
```

Additional References

The following sections provide references related to implementing tunnels.

Related Documents

Related Topic	Document Title
All Cisco IOS XE commands	Cisco IOS Master Command List, All Releases .

Related Topic	Document Title
Tunnel commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS Interface and Hardware Component Command Reference*
IPv6 commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IPv6 Command Reference*
Cisco IOS XE Interface and Hardware Component configuration modules	*Cisco IOS XE Interface and Hardware Component Configuration Guide*,
Cisco IOS XE IPv6 configuration modules	*Cisco IOS XE IPv6 Configuration Guide*,
Cisco IOS XE Quality of Service Solutions configuration modules	*Cisco IOS XE Quality of Service Solutions Configuration Guide*
Cisco IOS XE Multiprotocol Label Switching configuration modules	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*
Configuration example for a VRF-aware dynamic multipoint VPN (DMVPN)	"Dynamic Multipoint VPN (DMVPN)" configuration module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*

Standards/RFCs

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--
RFC 791	*Internet Protocol*
RFC 1191	*Path MTU Discovery*
RFC 1323	*TCP Extensions for High Performance*
RFC 1483	*Multiprotocol Encapsulation over ATM Adaptation Layer 5*
RFC 2003	*IP Encapsulation Within IP*
RFC 2018	*TCP Selective Acknowledgment Options*
RFC 2460	*Internet Protocol, Version 6 (IPv6)*
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*
RFC 2474	*Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
RFC 2516	*A Method for Transmitting PPP over Ethernet (PPPoE)*

Standard	Title
RFC 2547	*BGP/MPLS VPNs*
RFC 2780	*IANA Allocation Guidelines for Values in the Internet Protocol and Related Headers*
RFC 2784	*Generic Routing Encapsulation (GRE)*
RFC 2890	*Key and Sequence Number Extensions to GRE*
RFC 2893	*Transition Mechanisms for IPv6 Hosts and Routers*
RFC 3056	*Connection of IPv6 Domains via IPv4 Clouds*
RFC 3147	*Generic Routing Encapsulation over CLNS Networks*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Implementing Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 311: Feature Information for Implementing Tunnels

Feature Name	Releases	Feature Information
EoMPLS over GRE	Cisco IOS XE Release 2.5	The EoMPLS over GRE feature allows you to tunnel Layer 2 traffic through a Layer 3 MPLS network. This feature also helps to create the GRE tunnel as hardware-based switched, and with high performance that encapsulates EoMPLS frames within the GRE tunnel. No new commands were introduced or modified by this feature.

Feature Name	Releases	Feature Information
GRE Tunnel IP Source and Destination VRF Membership	Cisco IOS XE Release 2.2	The GRE Tunnel IP Source and Destination VRF Membership feature allows you to configure the source and destination of a tunnel to belong to any VPN VRF table. The following command was introduced or modified: **tunnel vrf**.
GRE Tunnel Keepalive	Cisco IOS XE Release 2.1	The GRE Tunnel Keepalive feature provides the capability of configuring keepalive packets to be sent over IP-encapsulated GRE tunnels. You can specify the rate at which keepalives will be sent and the number of times that a device will continue to send keepalive packets without a response before the interface becomes inactive. GRE keepalive packets may be sent from both sides of a tunnel or from just one side. The following command was introduced by this feature: **keepalive (tunnel interfaces)** .
IP over IPv6 Tunnels	Cisco IOS XE Release 2.4	The following commands were modified by this feature: **tunnel destination**, **tunnel mode**, and **tunnel source**.
IP Precedence for GRE Tunnels	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Aggregation Services Routers.
IP Tunnel— SSO	Cisco IOS XE Release 3.6	High availability support was added to IP Tunnels. No new commands were introduced or modified by this feature.
Tunnel ToS	Cisco IOS XE Release 2.1	The Tunnel ToS feature allows you to configure the ToS and Time-to-Live (TTL) byte values in the encapsulating IP header of tunnel packets for an IP tunnel interface on a router. The Tunnel ToS feature is supported in Cisco Express Forwarding, fast switching, and process switching forwarding modes. The following commands were introduced or modified by this feature: **show interfaces tunnel**, **tunnel tos**, **tunnel,** and **ttl**.

CHAPTER 257

Tunnel Route Selection

The Tunnel Route Selection feature allows the tunnel transport to be routed using a subset of the routing table. When there are equal-cost routes to a tunnel destination, normal tunnel transport behavior is to use one of the available routes chosen at random. The Tunnel Route Selection feature allows the explicit configuration of the outgoing interface for the tunnel transport.

Prerequisites for Tunnel Route Selection

Tunnel interfaces are configured.

Restrictions for Tunnel Route Selection

This feature is supported in the following tunnel modes only:

- Generic Routing Encapsulation (GRE) IP
- GRE Multipoint
- IP in IP
- Mobile User Datagram Protocol (UDP)

This feature is not supported on a tunnel when the tunnel transport is a GRE Multipoint tunnel.

Supported Configuration

```
interface tunnel 0
  tunnel mode gre multipoint
  tunnel route-via tunnel 1
```

```
interface tunnel 1
   tunnel mode gre ip
```

Unsupported Configuration

```
interface tunnel 0
  tunnel mode gre multipoint
  tunnel route-via tunnel 1
interface tunnel 1
  tunnel mode gre multipoint
```

Information About Tunnel Route Selection

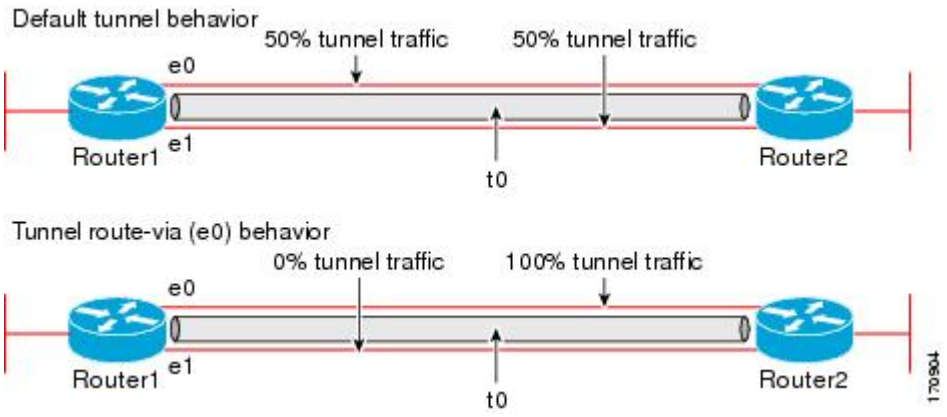
Tunnel Transport Behavior

The Tunnel Route Selection feature allows the tunnel transport to be routed using a subset of the routing table by specifying the outgoing interface of the tunnel transport.

The Tunnel Route Selection feature is not the same as an implementation of policy-based routing for the tunnel transport. The Tunnel Route Selection feature will forward traffic using only a subset of the route table, and it cannot introduce routing loops into the network.

The figure below compares default tunnel behavior with the Tunnel Route Selection behavior.

Figure 215: Tunnel Route Selection Traffic



How to Configure Tunnel Route Selection

Configuring Tunnel Route Selection

Perform the following steps to specify the outgoing interface of the tunnel transport to route the tunnel transport using a subset of the routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *interface-number*
4. **tunnel route-via** *interface-type* *interface-number* {**mandatory** | **preferred**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *interface-number* **Example:** `Router(config)# interface tunnel 0`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**tunnel route-via** *interface-type* *interface-number* {**mandatory** | **preferred**} **Example:** `Router(config-if)# tunnel route-via ethernet0 mandatory`	Specifies the outgoing interface to be used by the tunnel transport.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

To troubleshoot your configuration, use the **debug tunnel route-via** command in privileged EXEC mode. The following is sample output from the **debug tunnel route-via** command after the **tunnel route-via** command was used to route the tunnel transport explicitly using a subset of the routing table.

```
Router# debug tunnel route-via
Tunnel route-via debugging is on
Router#
*May 23 08:40:53.707: TUN-VIA: Tunnel0 candidate route-via Ethernet0/0, next hop 10.73.2.1
*May 23 08:40:53.707: TUN-VIA: Tunnel0 route-via action is forward
*May 23 08:41:03.719: TUN-VIA: Tunnel0 candidate route-via Ethernet0/0, next hop 10.73.2.1
```

```
*May 23 08:41:03.719: TUN-VIA: Tunnel0 route-via action is forward
Router# undebug tunnel route-via
Tunnel route-via debugging is off
```

What to Do Next

You can verify the tunnel route selection configuration. To verify your configuration, use the **show interfaces tunnel** command in privileged EXEC mode. The following example shows that the tunnel transport is routed using a subset of the routing table by specifying the outgoing interface of the tunnel transport.

```
Router# show running-config interface tunnel 0
Building configuration...
Current configuration : 147 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel source Loopback0
 tunnel destination 10.73.0.102
 tunnel route-via Ethernet0 preferred
end
Router# show interfaces tunnel 0 | include route-via
```

Tunnel route-via feature is on [Ethernet0, preferred]

Configuration Examples for Tunnel Route Selection

Example Configuring Tunnel Route Selection

The following example shows Tunnel 0 configured to use Ethernet interface 0 as its preferred outgoing transport interface. Traffic that exits the router using the tunnel 0 interface will be sent out of Ethernet interface 0 if there is a route to the tunnel destination out of Ethernet interface 0. If there is no route out of Ethernet interface 0, the traffic will be forwarded as if the Tunnel Route Selection feature were not configured.

If the **tunnel route-via** *interface-type interface-number* **mandatory** command is configured, and there is no route to the tunnel destination using that interface, a point-to-point tunnel interface will go into a down state.

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tunnel 0
Router(config-if)# tunnel route-via ethernet0 preferred
Router(config-if)# end
Router# show running-config interface tunnel 0
Building configuration...
Current configuration : 147 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel source Loopback0
 tunnel destination 10.73.0.102
 tunnel route-via Ethernet0 preferred
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Interface commands	Cisco IOS Interface and Hardware Component Command Reference
Configuration commands	Cisco IOS Configuration Fundamentals Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Tunnel Route Selection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 312: Feature Information for Tunnel Route Selection

Feature Name	Releases	Feature Information
Tunnel Route Selection	12.4(11)T 15.0(1)M Cisco IOS Release 3.9S	The Tunnel Route Selection feature allows the tunnel transport to be routed using a subset of the routing table. When there are equal-cost routes to a tunnel destination, normal tunnel transport behavior is to use one of the available routes chosen at random. The Tunnel Route Selection feature allows the explicit configuration of the outgoing interface for the tunnel transport. The following commands were introduced or modified: **debug tunnel route-via, tunnel route-via, show interfaces tunnel**.

CHAPTER 258

MPLS VPN over mGRE

The MPLS VPN over mGRE feature overcomes the requirement that a carrier support multiprotocol label switching (MPLS) by allowing you to provide MPLS connectivity between networks that are connected by IP-only networks. This allows MPLS label switched paths (LSPs) to use generic routing encapsulation (GRE) tunnels to cross routing areas, autonomous systems, and internet service providers (ISPs). when MPLS VPNs are configured over multipoint GRE (mGRE) you can deploy layer-3 (L3) provider edge (PE) based virtual private network (VPN) services using a standards-based IP core. This allows you to provision the VPN services without using the overlay method.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS VPN over mGRE

Before you configure MPLS VPN with mGRE tunnels, ensure that the MPLS VPN is configured and working properly. See the "Configuring MPLS Layer 3 VPNs" module for information about setting up MPLS VPNs.

Restrictions for MPLS VPN over mGRE

- Tunnelled tag traffic must enter the router through a line card that supports MPLS VPN over mGRE.
- Each PE router supports one tunnel configuration only.
- MPLS VPN over mGRE feature does not support transportation of multicast traffic between VPNs, however mVPN over Rosen based mGRE can co-exist with MPLS VPN over mGRE feature to provide solution for multicast VPN.
- When a GRE tunnel has the same destination address and source address as the mGRE, the tunnel gets route-cache switched.
- The packets that require fragmentation get route cache-switched.
- When an L3VPN profile is removed and added back, then you should clear the Border Gateway Protocol (BGP) using the **clear ip bgp soft**command.
- When an mGRE tunnel is created, a dummy tunnel is also created.
- The loopback or IP address used in the update source of the BGP configuration should be the same as that of the transport source of the L3VPN profile.
- mGRE is not stateful switchover (SSO) compliant. However, both mGRE and SSO coexist.
- You can configure mGRE and multicast distribution tree (MDT) tunnels with the same loopback address.

The limitations for MPLS VPN over mGRE feature are as follows:

-
  - Not all GRE options are supported in the hardware (for example, GRE extended header and GRE key).
  - Checking identical VLANs (Internet Control Message Protocol [ICMP] redirect) is not supported on the tunnels.
  - Features such as unicast reverse path forwarding (uRPF) and BGP policy accounting are not supported on the tunnels.

Information About MPLS VPN over mGRE

You can configure mGRE tunnels to create a multipoint tunnel network that overlays an IP backbone. This overlay connects PE routers to transport VPN traffic.

In addition, when MPLS VPNs are configured over mGRE you can deploy L3 PE-based VPN services using a standards-based IP core. This allows you to provision the VPN services without using the overlay method. When MPLS VPN over mGRE is configured, the system uses IPv4-based mGRE tunnels to encapsulate VPN-labeled IPv4 and IPv6 packets between PEs. To deploy MPLS VPN over mGRE tunnels, you create a VRF instance, enable and configure L3 VPN encapsulation, link the route map to the application template, and set up the BGP VPNv4 and VPNv6 exchange so that updates are filtered through the route map.

MPLS VPN over mGRE

GRE is a point-to-point tunneling protocol where two peers form the endpoints of the tunnel. It is designed to encapsulate network-layer packets inside IP tunneling packets. mGRE is a similar protocol with a single endpoint at one side of the tunnel connected to multiple endpoints at the other side of the tunnel. The mGRE tunnel provides a common link between branch offices that connect to the same VPN. Because mGRE is a point-to-multipoint model, fully meshed GRE tunnels are not required to interconnect MPLS VPN PE devices.

MPLS is a widely deployed VPN internet architecture. MPLS requires that all core routers in the network support MPLS. This feature is useful in networks where the service provider uses a backbone carrier to provide connectivity.

The MPLS VPN over mGRE feature overcomes the requirement of carrier support MPLS by allowing you to provide MPLS connectivity between networks that are connected by IP-only networks. This allows MPLS LSPs to use GRE tunnels to cross routing areas, autonomous systems, and ISPs.

When MPLS VPNs are configured over mGRE you can deploy L3 PE-based VPN services using a standards-based IP core. This allows you to provision the VPN services without using LSP or a Label Distribution Protocol (LDP). The system uses IPv4-based mGRE tunnels to encapsulate VPN-labeled IPv4 and IPv6 packets between PEs.

The MPLS VPN over mGRE feature also allows you to deploy existing MPLS VPN LSP-encapsulated technology concurrently with MPLS VPN over mGRE and enables the system to determine which encapsulation method is used to route specific traffic. The ingress PE router determines which encapsulation technology to use when a packet is sent to the remote PE router.

This section includes information on the following topics on MPLS VPN over mGRE feature:

Route Maps

By default, VPN traffic is sent using an LSP. The MPLS VPN over mGRE feature uses user-defined route maps to determine which VPN prefixes are reachable over an mGRE tunnel and which VPN prefixes are reachable using an LSP. The route map is applied to advertisements for VPNv4 and VPNv6 address families. The route map uses a next hop tunnel table to determine the encapsulation method for the VPN traffic.

To route traffic over the mGRE tunnel, the system creates an alternative address space that shows that all next hops are reached by encapsulating the traffic in an mGRE tunnel. To configure a specific route to use an mGRE tunnel, the user adds an entry for that route to the route map. The new entry remaps the Network Layer Reachability Information (NLRI) of the route to the alternative address space. If there is no remap entry in the route map for a route, then traffic on that route is forwarded over an LSP.

When the user configures MPLS VPN over mGRE, the system automatically provisions the alternative address space, normally held in the tunnel-encapsulated virtual routing and forwarding (VRF) instance. To ensure that all traffic reachable through the address space is encapsulated in an mGRE tunnel, the system installs a single default route out of a tunnel. The system also creates a default tunnel on the route map. The user can attach this default route map to the appropriate BGP updates.

Tunnel Endpoint Discovery and Forwarding

In order for the MPLS VPN over mGRE feature to function correctly, the system must be able to discover the remote PEs in the system and construct tunnel forwarding information for these remote PEs. In addition the system must be able to detect when a remote PE is no longer valid and remove the tunnel forwarding information for that PE.

If an ingress PE receives a VPN advertisement over BGP, it uses the route target attributes (which it inserts into the VRF) and the MPLS VPN label from the advertisement, to associate the prefixes with the appropriate customer. The next hop of the inserted route is set to the NLRI of the advertisement.

The advertised prefixes contain information about remote PEs in the system (in the form of NLRIs), and the PE uses this information to notify the system when an NLRI becomes active or inactive. The system uses this notification to update the PE forwarding information.

When the system receives notification of a new remote PE, it adds the information to the tunnel endpoint database, which causes the system to create an adjacency associated with the tunnel interface. The adjacency description includes information on the encapsulation and other processing that the system must perform to send encapsulated packets to the new remote PE.

The adjacency information is placed into the tunnel encapsulated VRF. When a user remaps a VPN NLRI to a route in the VRF (using the route map), the system links the NLRI to the adjacency; therefore the VPN is linked to a tunnel.

Tunnel Decapsulation

When the egress PE receives a packet from a tunnel interface that uses the MPLS VPN over mGRE feature, the PE decapsulates the packet to create a VPN label tagged packet, and sends the packet to the MPLS forwarding (MFI) code.

Tunnel Source

The MPLS VPN over mGRE feature uses a single tunnel configured as an mGRE tunnel to configure a system with a large number of endpoints (remote PEs). To identify the origin of tunnel-encapsulated packets, the system uses the tunnel source information.

At the transmitting (ingress) PE, when a VPN packet is sent to a tunnel, the tunnel destination is the NLRI. At a receiving (egress) PE, the tunnel source is the address that the packets encapsulated in the mGRE tunnel are received on. Therefore, at the egress PE the packet destination must match the NLRI from the local PE.

IPv6 VPN

If the advertising PE router has an IPv6 address then the NLRI must also be an IPv6 address (regardless of the network between the PEs). If the network between the PEs is IPv4 based, the system creates the IPv6 address of the advertising PE using an IPv4 mapped address in the following form: ::FFFF:IPv4-PE-address. The receiving PE sets the next hop for the VPN tag IPv6 prefixes to the IPv4 address embedded in the IPv6 NLRI. This enables the PE to link VPNv6 traffic to an LSP or an mGRE tunnel in the same way it maps VPNv4 traffic.

When a PE receives VPNv6 updates, it applies the IPv6 route map. The MPLS VPN over mGRE feature uses the IPv6 route map to set the next hop information in the Tunnel_Encap VRF.

How to Configure MPLS VPN over mGRE

Configuring an L3VPN Encapsulation Profile

This section describes how to configure an L3VPN encapsulation profile.

Note Transport protocols such as IPv6, MPLS, IP, and Layer 2 Tunneling Protocol version 3 (L2TPv3) can also be used in this configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l3vpn encapsulation ip** *profile-name*
4. **transport ipv4 [source** *interface-type interface-number* **]**
5. **protocol gre [key** *gre-key* **]**
6. **end**
7. **show l3vpn encapsulation ip** *profile-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**l3vpn encapsulation ip** *profile-name* **Example:** `Router(config)# l3vpn encapsulation ip tunnel encap`	Enters L3 VPN encapsulation configuration mode to create the tunnel.
Step 4	**transport ipv4 [source** *interface-type interface-number* **]** **Example:** `Router(config-l3vpn-encap-ip)# transport ipv4 source loopback 0`	(Optional) Specifies IPv4 transport source mode and defines the transport source interface. • If you use the **transport ipv4 source** *interface-type interface-number* command, make sure that the specified source address is used as the next hop in BGP updates advertised by the PE. • If you do not use this command, the **bgp update source**or **bgp next-hop** command is automatically used as the tunnel source.
Step 5	**protocol gre [key** *gre-key* **]** **Example:**	Specifies GRE as the tunnel mode and sets the GRE key.

	Command or Action	Purpose
	`Router(config-l3vpn-encap-ip)# protocol gre key 1234`	
Step 6	**end** **Example:** `Router(config-l3vpn-encap-ip)# end`	Exits L3 VPN encapsulation configuration mode and returns to privileged EXEC mode.
Step 7	**show l3vpn encapsulation ip** *profile-name* **Example:** `Router# show l3vpn encapsulation ip tunnel encap`	(Optional) Displays the profile health and the underlying tunnel interface.

Configuring BGP and Route Maps

Perform this task to configure BGP and route maps. The following steps also enable you to link the route map to the application template and set up the BGP VPNv4 and VPNv6 exchange so that the updates are filtered through the route map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** *ip-address* **remote-as** *as-number*
6. **neighbor** *ip-address* **update-source** *interface name*
7. **address-family ipv4**
8. **no synchronization**
9. **redistribute connected**
10. **neighbor** *ip-address* **activate**
11. **no auto-summary**
12. **exit**
13. **address-family vpnv4**
14. **neighbor** *ip-address* **activate**
15. **neighbor** *ip-address* **send-community both**
16. **neighbor** *ip-address* **route-map** *map-name* **in**
17. **exit**
18. **address-family vpnv6**
19. **neighbor** *ip-address* **activate**
20. **neighbor** *ip-address* **send-community both**
21. **neighbor** *ip-address* **route-map** *map-name* **in**
22. **exit**
23. **route-map** *map-tag* **permit** *position*
24. **set ip next-hop encapsulate l3vpn** *profile-name*

25. **set ipv6 next-hop encapsulate l3vpn** *profile-name*
26. **exit**
27. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Specifies the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along, and enters router configuration mode.
Step 4	**bgp log-neighbor-changes** **Example:** `Router(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor resets.
Step 5	**neighbor** *ip-address* **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 209.165.200.225 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 6	**neighbor** *ip-address* **update-source** *interface name* **Example:** `Router(config-router)# neighbor 209.165.200.225 update-source loopback 0`	Allows BGP sessions to use any operational interface for TCP connections.
Step 7	**address-family ipv4** **Example:** `Router(config-router)# address-family ipv4`	Enters address family configuration mode to configure routing sessions that use IPv4 address prefixes.
Step 8	**no synchronization** **Example:** `Router(config-router-af)# no synchronization`	Enables the Cisco software to advertise a network route without waiting for an IGP.

	Command or Action	Purpose
Step 9	**redistribute connected** **Example:** `Router(config-router-af)# redistribute connected`	Redistributes routes from one routing domain into another routing domain and allows the target protocol to redistribute routes learned by the source protocol and connected prefixes on those interfaces over which the source protocol is running.
Step 10	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 209.165.200.225 activate`	Enables the exchange of information with a BGP neighbor.
Step 11	**no auto-summary** **Example:** `Router(config-router-af)# no auto-summary`	Disables automatic summarization and sends subprefix routing information across classful network boundaries.
Step 12	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode.
Step 13	**address-family vpnv4** **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode to configure routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 14	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 209.165.200.225 activate`	Enables the exchange of information with a BGP neighbor.
Step 15	**neighbor** *ip-address* **send-community both** **Example:** `Router(config-router-af)# neighbor 209.165.200.225 send-community both`	Specifies that a communities attribute, for both standard and extended communities, should be sent to a BGP neighbor.
Step 16	**neighbor** *ip-address* **route-map** *map-name* **in** **Example:** `Router(config-router-af)# neighbor 209.165.200.225 route-map SELECT_UPDATE_FOR_L3VPN in`	Applies the named route map to the incoming route.
Step 17	**exit** **Example:**	Exits address family configuration mode.

	Command or Action	Purpose
	`Router(config-router-af)# exit`	
Step 18	**address-family vpnv6** **Example:** `Router(config-router)# address-family vpnv6`	Enters address family configuration mode to configure routing sessions, such as BGP, that use VPNv6 address prefixes.
Step 19	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 209.165.200.252 activate`	Enables the exchange of information with a BGP neighbor.
Step 20	**neighbor** *ip-address* **send-community both** **Example:** `Router(config-router-af)# neighbor 209.165.200.252 send-community both`	Specifies that a communities attribute, for both standard and extended communities, should be sent to a BGP neighbor.
Step 21	**neighbor** *ip-address* **route-map** *map-name* **in** **Example:** `Router(config-router-af)# neighbor 209.165.200.252 route-map SELECT_UPDATE_FOR_L3VPN in`	Applies the named route map to the incoming route.
Step 22	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode.
Step 23	**route-map** *map-tag* **permit** *position* **Example:** `Router(config-router)# route-map SELECT_UPDATE_FOR_L3VPN permit 10`	Enters route-map configuration mode and defines the conditions for redistributing routes from one routing protocol into another. • The **redistribute** router configuration command uses the specified map tag to reference this route map. Multiple route maps may share the same map tag name. • If the match criteria are met for this route map, the route is redistributed as controlled by the set actions. • If the match criteria are not met, the next route map with the same map tag is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set. • The *position* argument indicates the position a new route map will have in the list of route maps already configured with the same name.

	Command or Action	Purpose
Step 24	**set ip next-hop encapsulate l3vpn** *profile-name* **Example:** `Router(config-route-map)# set ip next-hop encapsulate l3vpn my profile`	Indicates that output IPv4 packets that pass a match clause of the route map are sent to the VRF for tunnel encapsulation.
Step 25	**set ipv6 next-hop encapsulate l3vpn** *profile-name* **Example:** `Router(config-route-map)# set ip next-hop encapsulate l3vpn tunnel encap`	Indicates that output IPv6 packets that pass a match clause of the route map are sent to the VRF for tunnel encapsulation.
Step 26	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 27	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Configuration Examples for MPLS VPN over mGRE

Example Verifying the MPLS VPN over mGRE Configuration

Use the following examples to verify that the configuration is working properly:

Cisco Express Forwarding (CEF) Switching

You can verify that CEF switching is working as expected:

```
Router# show ip cef vrf Customer_A tunnel 0

209.165.200.250
/24
    nexthop 209.165.200.251 Tunnel0 label 16
```

Endpoint Creation

You can verify the tunnel endpoint that has been created:

```
Router# show tunnel endpoints tunnel 0

 Tunnel0 running in multi-GRE/IP mode
 Endpoint transport 209.165.200.251 Refcount 3 Base 0x2AE93F0 Create Time 00:00:42
   overlay 209.165.200.254 Refcount 2 Parent 0x2AE93F0 Create Time 00:00:42
```

Adjacency

You can verify that the corresponding adjacency has been created:

```
Router# show adjacency tunnel 0
   Protocol Interface                 Address
  IP       Tunnel0                   209.165.200.251(4)
  TAG      Tunnel0                   209.165.200.251(3)
```

Profile Health

You can use **show l3vpn encapsulation** *profile-name* command to get information on the basic state of the application. The output of this command provides you details on the references to the underlying tunnel.

```
Router# show l3vpn encapsulation ip tunnel encap
 Profile: tunnel encap
 transport ipv4 source Auto: Loopback0
 protocol gre
   Tunnel Tunnel0 Created [OK]
   Tunnel Linestate [OK]
   Tunnel Transport Source (Auto) Loopback0 [OK]
```

Example Configuration Sequence for MPLS VPN over mGRE

This example shows the configuration sequence for MPLS VPN over mGRE:

```
vrf definition Customer A
 rd 100:110
 route-target export 100:1000
 route-target import 100:1000
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
!
ip cef
!
ipv6 unicast-routing
ipv6 cef
!
!
l3vpn encapsulation ip sample profile name
 transport source loopback 0
 protocol gre key 1234
!
!
 interface Loopback0
  ip address 209.165.200.252 255.255.255.224
  ip router isis
!
interface Serial2/0
 vrf forwarding Customer A
 ip address 209.165.200.253 255.255.255.224
 ipv6 address 3FFE:1001::/64 eui-64
 no fair-queue
 serial restart-delay 0
!
```

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 209.165.200.254 remote-as 100
 neighbor 209.165.200.254 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  redistribute connected
  neighbor 209.165.200.254 activate
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 209.165.200.254 activate
  neighbor 209.165.200.254 send-community both
  neighbor 209.165.200.254 route-map SELECT_UPDATE_FOR_L3VPN in
 exit-address-family
 !
 address-family vpnv6
  neighbor 209.165.200.254 activate
  neighbor 209.165.200.254 send-community both
  neighbor 209.165.200.254 route-map SELECT_UPDATE_FOR_L3VPN in
 exit-address-family
 !
 address-family ipv4 vrf Customer A
  no synchronization
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf Customer A
  redistribute connected
  no synchronization
 exit-address-family
!
!
route-map SELECT_UPDATE_FOR_L3VPN permit 10
set ip next-hop encapsulate sample profile name
set ipv6 next-hop encapsulate sample profile name
```

Additional References

Related Documents

Related Topic	Document Title
Configuring MPLS Layer 3 VPNs	*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*
Cisco Express Forwarding	*Cisco IOS XE IP Switching Configuration Guide*
Generic routing encapsulation	*Cisco IOS XE Interface and Hardware Component Configuration Guide*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
IETF-PPVPN-MPLS-VPN-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: *http://www.cisco.com/go/mibs*

RFCs

RFC	Title
RFC 2547	*BGP/MPLS VPNs*
RFC 2784	*Generic Routing Encapsulation (GRE)*
RFC 2890	*Key Sequence Number Extensions to GRE*
RFC 4023	Encapsulating MPLS in IP or Generic Routing Encapsulation
RFC 4364	*BGP/MPLS IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	*http://www.cisco.com/cisco/web/support/index.html*

Feature Information for MPLS VPN over mGRE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 313: Feature Information for MPLS VPN over mGRE

Feature Name	Releases	Feature Information
MPLS VPN over mGRE	Cisco IOS XE Release 3.1S	This feature provides support to carry MPLS Layer 3 VPN traffic over mGRE. The following commands were introduced or modified by this feature: l3vpn encapsulation ip, protocol gre, show l3vpn encapsulation ip, transport ipv4, set ip next-hop, set ipv6 next-hop.

CHAPTER 259

IP Tunnel MIBs

This module contains information about MIBs used with interfaces and hardware components. The IP Tunnel MIB feature provides a generic MIB for managing all IPv4- and IPv6-related tunnels, as outlined in RFC 4087, IP Tunnel MIB. Tunneling provides a way to encapsulate arbitrary packets inside a transport protocol. A number of tunneling mechanism s specified by Internet Engineering Task Force (IETF) are implemented by Cisco for both IPv4 and IPv6 environments. Various MIBs are available for managing tunnels.

Prerequisites for the IP Tunnel MIB

Configure Simple Network Management Protocol (SNMP) on the router on which the IP Tunnel MIB feature is to be used. See the Configuring the Router to Use SNMP, on page 3113 for more information. For more information on configuring an SNMP server, see the "Configuring SNMP Support " chapter of the Cisco IOS Network Management Configuration Guide.

Restrictions for the IP Tunnel MIB

The IP Tunnel MIB feature supports only tunnels that can be created using the **interface tunnel** command. The IP Tunnel MIB feature does not support Layer 2 Tunnel Protocol (L2TP), Point-to-Point Tunneling Protocol (PPTP), or Multiprotocol Label Switching (MPLS) tunnels.

Information About the IP Tunnel MIB

Benefits of the IP Tunnel MIB

Improved Quality of Networks

Better IP tunnel instrumentation leads to an improvement in the quality of networks and better service delivery. A better quality network allows service providers to deliver a more reliable service.

Increased Reliability

The IP Tunnel MIB allows users of network management systems to set inventory and receive notification about their IP tunnel activity.

The IP Tunnel MIB supports both IPv4 and IPv6 network layers as defined in RFC 3291, and is used to manage IP tunnels implemented in the Cisco IOS software.

The IP Tunnel MIB supports all tunnel types, as well as tunnel creation and destruction capability.

Interoperability with Devices Other Than Cisco Devices

The IP Tunnel MIB works with key network management systems, including those of third-party vendors.

MIB Objects Supported by the IP Tunnel MIB

The following MIB objects are supported by the IP Tunnel MIB feature. For details regarding use of MIB objects, see RFC 4087, IP Tunnel MIB.

Table 314: Objects Supported by the IP Tunnel MIB

MIB Object	Description
tunnelIfEntry	Contains information on a particular configured tunnel. You can use the **interface tunnel** command to set a value for this object.
tunnelIfEncapsMethod	The encapsulation method used by the tunnel. You can use the **tunnel mode** command to set a value for this object.
tunnelIfHopLimit	Defines the IPv4 time to live (TTL) or IPv6 hop limit to use in the outer IP header. You can use the **tunnel ttl** command to set a value for this object.
tunnelIfSecurity	Used by the tunnel to secure the outer IP header. The value ipsec indicates that IPsec is used between the tunnel endpoints for authentication or encryption, or both.
tunnelIfTOS	Used by the tunnel to set the high 6 bits (the differentiated services codepoint) of the IPv4 type of service (ToS) or IPv6 traffic class in the outer IP header. You can use the **tunnel tos** command to set a value for this object.
tunnelIfFlowLabel	Used to set the IPv6 Flow Label value. This object is supported for tunnels over IPv6. The default value for this object is 0.

MIB Object	Description
tunnelIfAddressType	Shows the type of address in the corresponding tunelIfLocalInetAddress and tunnelIfRemoteInetAddress objects. This object cannot be configured individually through the command-line interface (CLI).
tunnelIfLocalInetAddress	The address of the local endpoint of the tunnel (that is, the source address used in the outer IP header). If the address is unknown, the value is 0.0.0.0 for IPv4 or :: for IPv6. The address type of this object is given by tunnelIfAddressType. You can use the **tunnel source** command to set a value for this object.
tunnelIfRemoteInetAddress	The address of the remote endpoint of the tunnel (that is, the destination address used in the outer IP header). If the address is unknown or the tunnel is not a point-to-point link (for example, a 6-to-4 tunnel), the value is 0.0.0.0 for tunnels over IPv4 or :: for tunnels over IPv6. The address type of this object is given by tunnelIfAddressType. You can use the **tunnel destination** command to set a value for this object.
tunnelIfEncapsLimit	Shows the maximum number of additional encapsulations permitted for packets undergoing encapsulation at this node. A value of -1 indicates that no limit is present (except as result of packet size).
tunnelInetConfigEntry	Contains information on a particular configured tunnel. There will be only one entry for multipoint tunnels and for tunnels that have the remote inet address 0.0.0.0 for IPv4 or :: for IPv6. Only generic routing encapsulation (GRE)/IP and GRE/IPv6 tunnels are created through the MIB.
tunnelInetConfigIfIndex	Shows the value of ifIndex corresponding to the tunnel interface. A value of 0 is not legal in the active state and means that the interface index has not yet been assigned.
tunnelInetConfigStatus	Used to create or delete table entries in the MIB table. You can use the **interface tunnel** to set a value for this object.
tunnelInetConfigStorageType	Indicates the storage type. Only a nonvolatile storage value is supported.

How to Configure SNMP and Use the IP Tunnel MIB

Configuring the Router to Use SNMP

Note Some of the tasks in this section include examples of the SNMP CLI syntax used to set configuration parameters on the router and to read values from MIB objects on the router. These SNMP CLI syntax examples are taken from a Linux workstation using public domain SNMP tools. The SNMP CLI syntax for your workstation might be different. See the documentation that was provided with your SNMP tools for the correct syntax for your network management workstation.

Before you can use the IP Tunnel MIB feature, you must first configure the router to support SNMP. Perform this task to enable SNMP on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string1* **ro**
4. **snmp-server community** *string2* **rw**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *string1* **ro** **Example:** `Router(config)# snmp-server community public ro`	Sets up the community access string to permit access to SNMP. • The *string1* argument is a community string that consists of from 1 to 32 alphanumeric characters and functions much like a password, permitting access to the SNMP protocol. Blank spaces are not permitted in the community string. • The **ro** keyword specifies read-only access. SNMP management stations using this string can retrieve MIB objects. **Note** The SNMP community read-only (RO) string for the examples is public. You should use a more complex string for this value in your configuration.
Step 4	**snmp-server community** *string2* **rw** **Example:** `Router(config)# snmp-server community private rw`	Sets up the community access string to permit access to SNMP. • The *string2* argument is a community string that consists of from 1 to 32 alphanumeric characters and functions much like a password, permitting access to the SNMP protocol. Blank spaces are not permitted in the community string. • The **rw** keyword specifies read-write access. SNMP management stations using this string can retrieve and modify MIB objects.

	Command or Action	Purpose
		Note The SNMP community read-write (RW) string for the examples is private. You should use a more complex string for this value in your configuration.
Step 5	**end** **Example:** `Router(config)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

What to Do Next

To implement the IP Tunnel MIB, you must configure a tunnel. For information on configuring tunnels, see the " Implementing Tunnels " chapter in the Cisco IOS Interface and Hardware Component Configuration Guide.

To debug or troubleshoot any issues related to configuring the IP Tunnel MIB through SNMP, use the debug snmp tunnel-mib command. For information on this command see Cisco IOS Interface and Hardware Component Command Reference.

Additional References

Related Documents

Related Topic	Document Title
SNMP commands, complete command syntax, command reference, command history, defaults, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
Configuring SNMP Support	*Cisco IOS Network Management Configuration Guide*
Implementing tunnels	*Cisco IOS Interface and Hardware Component Configuration Guide*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
IP Tunnel MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: *http://www.cisco.com/go/mibs*

RFCs

RFC	Title
RFC 4087	IP Tunnel MIB

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	*http://www.cisco.com/cisco/web/support/index.html*

Feature Information for the Tunnel MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 315: Feature Information for the IP Tunnel MIB

Feature Name	Releases	Feature Information
IP Tunnel MIB	12.2(33)SRB 12.2(1st)SY 12.2(44)SG 12.2(33)SRD 15.0(1)M Cisco IOS XE 3.1.0SG Cisco IOS XE Release 3.9S	The IP Tunnel MIB provides a generic MIB for managing all IPv4- and IPv6-related tunnels, as outlined in RFC 4087 IP Tunnel MIB.

CHAPTER 260

Synchronous Ethernet (SyncE) ESMC and SSM

This module describes Synchronization Status Message (SSM), Ethernet Synchronization Message Channel (ESMC), and generating the Simple Network Management Protocol (SNMP) traps on the SyncE feature.

With Ethernet equipment gradually replacing Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) equipment in service-provider networks, frequency synchronization is required to provide high-quality clock synchronization over Ethernet ports.

Synchronous Ethernet (SyncE) provides the required synchronization at the physical level. In SyncE, Ethernet links are synchronized by timing their bit clocks from high-quality, stratum-1-traceable clock signals in the same manner as SONET/SDH. Operation messages maintain SyncE links and ensure that a node always derives timing from the most reliable source.

SyncE synchronizes clock frequency over an Ethernet port. In SONET/SDH the communication channel for conveying clock information is SSM, and in SyncE it is the ESMC.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Synchronous Ethernet (SyncE) ESMC and SSM

You need to first configure the network clock for SyncE configuration. Automatic synchronization of the network clock should be enabled. Ensure that the **network-clock-select** and **network-clock-participate** commands do not exist in the configuration in order to continue with the SyncE configuration.

Restrictions for Synchronous Ethernet (SyncE) ESMC and SSM

- The **network-clock synchronization ssm option** command cannot be used if the following parameters have been configured:
  - Network clock input source using the **network-clock input-source** command.
  - Network clock quality level using the **network-clock quality-level** command.
  - Network clock source quality for any synchronous ethernet interface using the **network-clock source quality** command.

Note After using the **network-clock synchronization ssm option** command, the restricted configurations listed above can be used.

- The **network-clock synchronization ssm option** command must be compatible with the **network-clock eec** command in the configuration.
- The **esmc process** and **synchronous mode** commands can be used only if the SyncE capable interface is installed on the router.

Information About Synchronous Ethernet (SyncE) ESMC and SSM

Synchronous Ethernet (SyncE) ESMC and SSM

Customers using a packet network find it difficult to provide timing to multiple remote network elements (NEs) through an external time division multiplexed (TDM) circuit. The SyncE feature helps to overcome this problem by providing effective timing to the remote NEs through a packet network. SyncE leverages the physical layer of the Ethernet to transmit frequency to the remote sites. SyncE's functionality and accuracy resemble the SONET/SDH network because of its physical layer characteristic. SyncE uses ESMC to allow the best clock source traceability to correctly define the timing source and help prevent a timing loop.

SONET/SDH use 4 bits from the two S bytes in the SONET/SDH overhead frame for message transmission. Ethernet relies on ESMC that is based on an IEEE 802.3 organization-specific slow protocol for message transmission. Each NE along the synchronization path supports SyncE, and SyncE effectively delivers frequency in the path. SyncE does not support relative time (for example, phase alignment) or absolute time (Time of Day).

SyncE provides the Ethernet physical layer network (ETY) level frequency distribution of known common precision frequency references. Clocks for use in SyncE are compatible with the clocks used in the SONET/SDH synchronization network. To achieve network synchronization, synchronization information is transmitted through the network via synchronous network connections with performance of egress clock. In SONET/SDH the communication channel for conveying clock information is Synchronization Status Message (SSM), and in SyncE it the Ethernet Synchronization Message Channel (ESMC).

ESMC carries a Quality Level (QL) identifier that identifies the timing quality of the synchronization trail. QL values in QL-TLV are the same as QL values defined for SONET and SDH SSM. Information provided by SSM QLs during the network transmission helps a node derive timing from the most reliable source and prevents timing loops. ESMC is used with the synchronization selection algorithms. Because Ethernet networks are not required to be synchronous on all links or in all locations, the ESMC channel provides this service. ESMC is composed of the standard Ethernet header for an organization-specific slow protocol; the ITU-T OUI, a specific ITU-T subtype; an ESMC-specific header; a flag field; and a type, length, value (TLV) structure. The use of flags and TLVs improves the management of SyncE links and the associated timing change.

How to Configure Synchronous Ethernet (SyncE) ESMC and SSM

Configuring SyncE

Perform this task to configure SyncE using ESMC and SSM.

SUMMARY STEPS

1. **enable**
2. **network-clock set l ockout** {**external** *slot* / *card* / *port*[**10m**| **2m**| **t1**{**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*}
3. **network-clock clear lockout** {**external** *slot* / *card* / *port* [**10m**| **2m**| **t1** {**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*}
4. **network-clock switch force { external** *slot* / *card* / *port* [**10m** | **2m**] | **t0** | **t1** {**sf** | **esf** | **d4**} **t0 | internal { external** *slot* / *card* / *port*[**10m** | **2m**] **| t0} | interface** *type slot* / *port* **externa**l *slot* / *card* / *port* [**10m** | **2m**] | **t0** }
5. **network-clock switch manual { interface** *type slot* /port *{* **externa**l *slot* / *card* / *port* [**10m** | **2m**] | **t0** } | **external** *slot* / *card* / *port*{**10m** | **2m** | **t0** | **t1** {**sf** | **esf** | **d4**} | **internal { external** *slot* / *card* / *port*[**10m** | **2m**] | **t0}** *}*
6. **network-clock clear switch** {**t0** | **external** *slot* / *card* / *port* [**10m** | **2m**]}
7. **configure terminal**
8. **network-clock synchronization automatic**
9. **network-clock synchronization ssm option** {**1**| **2**{**GEN1**| **GEN2**}}
10. **network-clock input-source** *priority* {**external** *slot* / *card* / *port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*}
11. **network-clock synchronization mode ql-enabled**
12. **network-clock hold-off** {**0**| *milliseconds*}
13. **network-clock wait-to-restore** *seconds*
14. **esmc process**
15. **network-clock external** *slot* / *card* / *port* **hold-off** {**0** | *milliseconds*}

16. **network-clock quality-level** {**tx**| **rx**} *value* {**interface** *type slot / port* | **external** *slot / card / port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}]
17. **network-clock output-source {line** | **system**} *priority interface type slot / port* **external** *slot / card / port*[**10m** | **2m** | **t1**{**sf** | **esf** | **d4**} **]**
18. **interface** *type number*
19. **synchronous mode**
20. **esmc mode** [**ql-disabled**| **tx**| **rx**] *value*
21. **network-clock source quality-level** *value* {**tx** | **rx**}
22. **network-clock hold-off** {**0** | *milliseconds*}
23. **network-clock wait-to-restore** *seconds*
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**network-clock set l ockout** {**external** *slot / card / port*[**10m**	**2m**	**t1**{**sf**	**esf**	**d4**}]	**interface** *type slot / port*} **Example:** `Router# network-clock set lockout GigabitEthernet7/1`	Sets the lockout state of input to "on." The input then is no longer considered available by the selection process.
Step 3	**network-clock clear lockout** {**external** *slot / card / port* [**10m**	**2m**	**t1** {**sf**	**esf**	**d4**}]	**interface** *type slot / port*} **Example:** `Router# network-clock clear lockout GigabitEthernet7/1`	Sets the lockout state of input to "off." The input then is considered available by the selection process.
Step 4	**network-clock switch force { external** *slot / card / port* [**10m**	**2m**]	**t0**	**t1** {**sf**	**esf**	**d4**} **t0	internal { external** *slot / card / port*[**10m**	**2m**] **| t0}	interface** *type slot / port* **externa**l *slot / card / port* [**10m**	**2m**]	**t0** } **Example:** `Router# network-clock switch force interface GigabitEthernet 7/1 t0`	Overrides the currently selected synchronization source when the synchronization source is enabled and not locked out. If the source selected by the forced switch command is disabled or locked out, the forced switch command is automatically rejected.
Step 5	**network-clock switch manual { interface** *type slot* /port { **externa**l *slot / card / port* [**10m**	**2m**]	**t0** }	**external** *slot / card / port*{**10m**	**2m**	**t0**	**t1** {**sf**	**esf**	**d4**}	**internal { external** *slot / card / port*[**10m**	**2m**] **| t0}** *}* **Example:**	Selects the synchronization source interface when it is enabled and not locked out. Manual switching is used to override the previously assigned synchronization source priorities.

	Command or Action	Purpose
	`Router# network-clock switch manual interface GigabitEthernet 7/1 t0`	
Step 6	**network-clock clear switch** {**t0** | **external** *slot* / *card* / *port* [**10m** | **2m**]} **Example:** `Router# network-clock clear switch t0`	Clears the forced switch and manual switch commands. If the interface is not specified, the force/manual selected interface gets automatically cleared.
Step 7	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 8	**network-clock synchronization automatic** **Example:** `Router(config)# network-clock synchronization automatic`	Enables the network clock selection algorithm. This command disables the Cisco-specific network clock process and turns on the G.781-based automatic clock selection process.
Step 9	**network-clock synchronization ssm option** {**1**| **2**{**GEN1**| **GEN2**}} **Example:** `Router(config)# network-clock synchronization ssm option 2 GEN2`	Configures the router to work in a synchronization network. • Option 1 refers to synchronization networks designed for Europe. This is the default value. • Option 2 refers to synchronization networks designed for United States.
Step 10	**network-clock input-source** *priority* {**external** *slot* / *card* / *port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*} **Example:** `Router(config)# network-clock input-source 1 interface GigabitEthernet 7/1`	Enables selecting an interface that is configured as clock source line, an external timing input interface, a GPS interface, or a packet-based timing recovered clock as the input clock for the system. Interface can be SyncE or channelized SONET.
Step 11	**network-clock synchronization mode ql-enabled** **Example:** `Router(config)# network-clock synchronization mode ql-enabled`	Configures the automatic selection process ql-enabled mode. • QL is disabled by default. • ql-enabled mode can be used only when the synchronization interface is capable to send SSM.
Step 12	**network-clock hold-off** {**0**| *milliseconds*} **Example:** `Router(config)# network-clock hold-off 0`	(Optional) Configures hold-off timer for the interface.

	Command or Action	Purpose
Step 13	**network-clock wait-to-restore** *seconds* **Example:** `Router(config)# network-clock wait-to-restore 70`	(Optional) Configures wait-to-restore timer for the SyncE interface.
Step 14	**esmc process** **Example:** `Router(config)# esmc process`	Enables the ESMC process.
Step 15	**network-clock external** *slot* **/** *card* **/** *port* **hold-off** {**0** | *milliseconds*} **Example:** `Router(config)# network-clock external 0/1/0 hold-off 0`	Overrides the hold-off timer value for the external interface.
Step 16	**network-clock quality-level** {**tx**| **rx**} *value* {**interface** *type slot* **/** *port* | **external** *slot* **/** *card* **/** *port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}] **Example:** `Router(config)# network-clock quality-level rx QL-STU GigabitEthernet 0/0/0`	Forces the QL value for line or external timing input and output.
Step 17	**network-clock output-source {line** | **system**} *priority interface type slot* **/** *port* **external** *slot* **/** *card* **/** *port*[**10m** | **2m** | **t1**{**sf** | **esf** | **d4**} **]** **Example:** `Router(config)# network-clock output-source line 1 GigabitEthernet1/2 external 0/0/1 10m`	Transmits the signal from the external timing input interface to the external timing output interface.
Step 18	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0`	Enters interface configuration mode.
Step 19	**synchronous mode** **Example:** `Router(config-if)# synchronous mode`	Configures the Ethernet interface to synchronous mode and automatically enables the ESMC and QL process on the interface.
Step 20	**esmc mode** [**ql-disabled**| **tx**| **rx**] *value* **Example:** `Router(config-if)# esmc mode rx QL-STU`	(Optional) Enables the ESMC process on the interface.

	Command or Action	Purpose
Step 21	**network-clock source quality-level** *value* {**tx** | **rx**} **Example:** `Router(config-if)# network-clock source quality-level QL-ST4 tx`	(Optional) Provides the forced QL value to the local clock selection process.
Step 22	**network-clock hold-off** {**0** | *milliseconds*} **Example:** `Router(config-if)# network-clock hold-off 0`	(Optional) Configures the hold-off timer for the interface.
Step 23	**network-clock wait-to-restore** *seconds* **Example:** **Example:** `Router(config-if)# network-clock wait-to-restore 70`	(Optional) Configures the wait-to-restore timer for the SyncE interface.
Step 24	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling and Disabling an SNMP Trap in the SyncE Event

A Simple Network Management Protocol (SNMP) trap is defined for an SNMP agent to notify the Network Management Systems (NMS) about any unsolicited information. The SNMP trap notifies NMS when a critical SyncE event occurs on a device. If the SNMP trap is enabled in the SyncE configuration, the SNMP agent code generates a SyncE trap for the SyncE events.

Perform the following tasks to enable and disable the SNMP trap for the SyncE event:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps netsync**
4. **no snmp-server enable traps netsync**
5. **end**
6. **show running-config all | include traps**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps netsync** **Example:** `Router(config)# snmp-server enable traps netsync`	Enables the SyncE traps.
Step 4	**no snmp-server enable traps netsync** **Example:** `Router(config)# no snmp-server enable traps netsync`	(Optional) Disables the SyncE traps.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode.
Step 6	**show running-config all | include traps** **Example:** `Router# show running-config all | include trap`	(Optional) Displays the SyncE traps that are enabled on the router.

Configuration Examples for Synchronous Ethernet (SyncE) ESMC and SSM

Example Synchronous Ethernet (SyncE) ESMC and SSM

The following examples shows the SyncE configuration sequence (configuring an interface with two SyncE interfaces and two external interfaces):

```
Interface GigabitEthernet0/0/0
    synchronous mode
    clock source line
    network-clock wait-to-restore 720
!
Interface GigabitEthernet1/0/0
    synchronous mode
    clock source line
```

```
!
network-clock synchronization automatic
network-clock input-source 1 external 0/0/0 2m
network-clock input-source 2 external 1/0/0 2m
network-clock output-source line 1 interface GigabitEthernet0/0/0 external 0/0/0 2m
network-clock output-source line 1 interface GigabitEthernet1/0/0 external 1/0/0 2m
```

The following examples shows how to verify whether ESMC is enabled or not:

```
Router# show esmc

Interface: GigabitEthernet0/0/0
Administrative configurations:
  Mode: Synchronous
  ESMC TX: Enable
  ESMC RX : Enable
  QL RX configured : NA
  QL TX configured : NA
Operational status:
  Port status: UP
  QL Receive: QL-SSU-B
  ESMC Information rate : 1 packet/second
  ESMC Expiry: 5 second
```

The following examples shows how to view the network clock synchronization details:

```
Router# show network-clock synchronization detail

Automatic selection process : Enable
Equipment Clock : 2048 (EEC-Option1)
Clock Mode : QL-Enable
ESMC : Disabled
SSM Option : 1
T0 : Internal
Hold-off (global) : 300 ms
Wait-to-restore (global) : 300 sec
Revertive : No
Force Switch: FALSE
Manual Switch: FALSE
Number of synchronization sources: 1
Secondary src: Ethernet0/0
Slots disabled 0x0
Monitor source(s):  Ethernet0/0
Selected QL: QL-SEC
sm(netsync_ql_dis NETCLK_QL_ENABLE), running yes, state 1A
Last transition recorded: (begin)-> 1A (ql_mode_enable)-> 1A (src_added)-> 1A


Nominated Interfaces

 Interface            SigType     Mode/QL      Prio  QL_IN     ESMC Tx   ESMC Rx
*Internal             NA          NA/Dis       251   QL-SEC    NA        NA
 Et0/0                NA          Sync/En      2     QL-DNU    -         -

Interface:
-----------------------------------------------
Local Interface: Internal
Signal Type: NA
Mode: NA(Ql-enabled)
SSM Tx: Disable
SSM Rx: Disable
Priority: 251
```

```
QL Receive: QL-SEC
QL Receive Configured: -
QL Receive Overrided: -
QL Transmit: -
QL Transmit Configured: -
Hold-off: 0
Wait-to-restore: 0
Lock Out: FALSE
Signal Fail: FALSE
Alarms: FALSE
Slot Disabled: FALSE

Local Interface: Et0/0
Signal Type: NA
Mode: Synchronous(Ql-enabled)
ESMC Tx: Enable
ESMC Rx: Enable
Priority: 2
QL Receive: QL-DNU
QL Receive Configured: -
QL Receive Overrided: -
QL Transmit: -
QL Transmit Configured: -
Hold-off: 300
Wait-to-restore: 300
Lock Out: FALSE
Signal Fail: FALSE
Alarms: FALSE
Slot Disabled: FALSE
Dont Use: FALSE
Configured Priority: 2
Force Switch: FALSE
Manual Switch: FALSE
Manual Switch In progress: FALSE
Holdoff_cfg: FALSE
Wtr_cfg: FALSE
Reason for alarm flag: 0
Msw in progress: FALSE
Intf_sig_nv: 0
Hold off Timer: Stopped
Wait to restore Timer: Stopped
Switchover Timer: Stopped
ESMC Tx Timer: Stopped
ESMC Rx Timer: Stopped
Tsm Delay Timer: Stopped
```

Example Enabling and Disabling an SNMP Trap in the SyncE Event

The following example shows how to enable and disable an SNMP trap in the SyncE event:

```
Router > enable
Router # configure terminal
Router(config)# snmp-server enable traps netsync
Router (config)# no snmp-server enable traps netsync
Router (config)# end
Router# show running-config all| include traps
snmp-server enable traps flowmon
snmp-server enable traps sonet
snmp-server enable traps netsync
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Interface and hardware component configuration commands	*Cisco IOS Interface and Hardware Component Command Reference*

Standards

Standard	Title
ITU-T G.8262	*Timing characteristics of synchronous ethernet equipment slave clock (EEC)*
ITU-T G.8264	*Timing distribution through Packet Networks*
ITU-T G.781	*Synchronization layer functions*

MIBs

MIB	MIBs Link
CISCO-NETSYNC-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Synchronous Ethernet (SyncE) ESMC and SSM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 316: Feature Information for Synchronous Ethernet (SyncE): ESMC and SSM

Feature Name	Releases	Feature Information
Generating SNMP Trap in SyncE Feature	15.1(2)S Cisco IOS XE Release 3.8S	This feature describes how to set SNMP traps in SyncE to notifies the NMS about any unsolicited information. The following commands were introduced or modified by this feature: **no snmp-server enable traps netsync, show running-config all| include trap, snmp-server enable traps netsync.**
Synchronous Ethernet (SyncE): ESMC and SSM	15.0(1)S Cisco IOS XE Release 3.8S	This feature supports ESMC and the SSM control protocol for SyncE to synchronize clock frequency over an Ethernet port with quality level selection. The following commands were introduced or modified by this feature: **esmc mode ql-disabled**, **esmc process**, **show esmc**, **show interfaces accounting**.

CHAPTER 261

1+1 SR-APS Without Bridging

The Automatic Protection Switching (APS) feature provides link redundancy and allows switchover of Packet over SONET (POS) circuits in the event of circuit failure and is often required when you connect Synchronous Optical Networking (SONET) equipment to telecommunications equipment. In the single router (SR) APS feature both protect and working interfaces must be on same router.

APS is a mechanism of using a protect POS interface in the SONET network as the backup for a working POS interface. When the working interface fails, the protect interface quickly assumes its traffic load. Based on the configuration, the two circuits can be terminated in the same router. The protection mechanism has a 1+1 architecture with bidirectional connection. Bridging refers to the transmission of user data to both working interface and protect interface. In nonbridging scenario the user data is sent to working interface only.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for 1+1 SR-APS Without Bridging

Configure the working interface first, along with the IP address of the interface. This configuration helps to prevent the protect interface from becoming the active circuit during APS configuration. If the protect interface

becomes active in case if it has been configured first by mistake, you can use **shut** or **no shut** command to make the working interface active.

Restrictions for 1+1 SR-APS Without Bridging

- Both the protect and working interfaces should be configured identically. No warning message will be displayed if the configurations are different between the interfaces.
- Behavior of the APS pair (protect and working interfaces) will be indeterministic if the configurations of protect and working interfaces are not identical.
- APS switch over within 50 milliseconds is not supported during online insertion and removal (OIR) or during crash of the shared port adapter (SPA) or carrier card (CC).
- APS switching simultaneously with Route Processor (RP) or forwarding plane (FP) high availability (HA) need not be within 50 milliseconds.

Information About 1+1 SR-APS Without Bridging

1+1 SR-APS Without Bridging

The APS feature provides link redundancy and allows switchover of POS circuits in the event of circuit failure and is often required when you connect SONET equipment to telecommunications equipment. In the SR-APS feature both protect and working interfaces must be on same router.

APS is a mechanism of using a protect POS interface in the SONET network as the backup for a working POS interface. When the working interface fails, the protect interface quickly assumes its traffic load. Based on the configuration, the two circuits can be terminated in the same router. The protection mechanism has a 1+1 architecture with bidirectional connection.

In the 1+1 architecture, there is one working interface (circuit) and one protect interface, and the same payload from the transmitting end is sent to both the receiving ends. The receiving end decides the interface that needs to be used. The line overhead (LOH) bytes (K1 and K2) in the SONET frame indicate both status and action. When one interface is down or the K1/K2 bytes have changed, APS brings up the protect interface using regular interface configuration messages.

Bridging refers to the transmission of user data to both the working interface and the protect interface. In nonbridging scenario the user data is sent to the working interface only. You must set the working interface to be the active interface.

SR-APS uses Protect Group Protocol (PGP) between working and protect interfaces. The protect interface APS configuration should include an IP address of a loopback interface on the same router to communicate with the working interface using PGP. Using the PGP, POS interfaces can be switched in case of a degradation or loss of channel signal, or manual intervention. In bidirectional mode, the receive and transmit channels are switched as a pair.

In bidirectional APS the local and the remote connections negotiate the ingress interface to be selected for the data path. The egress interface traffic is not transmitted to both working and protect interfaces.

How to Configure 1+1 SR-APS Without Bridging

Configuring APS Working and Protect Interfaces

Perform this task to configure APS working and protect interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **aps working** *circuit-number*
5. **aps protect** *circuit-number ip-address*
6. **end**
7. **show controllers pos**
8. **show interfaces pos**
9. **show aps**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface pos** *slot/sub-slot/port* **Example:** `Router(config)# interface pos 2/0/0`	Specifies the POS interface to be configured as the working interface and enters interface configuration mode.
Step 4	**aps working** *circuit-number* **Example:** `Router(config-if)# aps working 1`	Configures the interface as a working interface.
Step 5	**aps protect** *circuit-number ip-address* **Example:** `Router(config-if)# aps protect 1 209.165.200.224`	Configures the interface as a protect interface. Specifies the IP address of loopback interface on the same router that contains the working interface.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show controllers pos** **Example:** `Router(config)# show controllers pos`	Displays information about the POS controllers so that you can verify that the interface is configured correctly.
Step 8	**show interfaces pos** **Example:** `Router(config)# show interfaces pos`	Displays information about the configured interfaces.
Step 9	**show aps** **Example:** `Router(config)# show aps`	Displays information about APS on the configured router.

Configuring Other APS Options

Perform this task to configure other APS options.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **aps force** *circuit-number*
5. **aps group** *group-number*
6. **aps lockout** *circuit-number*
7. **aps manual** *circuit-number*
8. **aps revert** *minutes*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface pos** *slot/sub-slot/port* **Example:** `Router(config)# interface pos 2/0/0`	Specifies the POS interface to be configured as the working interface and enters interface configuration mode.
Step 4	**aps force** *circuit-number* **Example:** `Router(config-if)# aps force 1`	(Optional) Manually switches the specified circuit to a protect interface, unless a request of equal or higher priority is in effect.
Step 5	**aps group** *group-number* **Example:** `Router(config-if)# aps group 20`	(Optional) Allows more than one protect or working interface group to be supported on a router.
Step 6	**aps lockout** *circuit-number* **Example:** `Router(config-if)# aps lockout 1`	(Optional) Prevents a working interface from switching to a protect interface.
Step 7	**aps manual** *circuit-number* **Example:** `Router(config-if)# aps manual 1`	(Optional) Manually switches a circuit to a protect interface, unless a request of equal or higher priority is in effect.
Step 8	**aps revert** *minutes* **Example:** `Router(config-if)# aps revert 3`	(Optional) Enables automatic switchover from the protect interface to the working interface after the working interface becomes available.
Step 9	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining APS

Perform this task to monitor and maintain APS.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **show controllers pos**
4. **show interfaces pos**
5. **show aps**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**show controllers pos** **Example:** `Router(config)# show controllers pos`	Displays information about the POS controllers so that you can verify that the interface is configured correctly.
Step 4	**show interfaces pos** **Example:** `Router(config)# show interfaces pos`	Displays information about the configured interfaces.
Step 5	**show aps** **Example:** `Router(config)# show aps`	Displays information about APS on the configured router.

Configuring SONET Alarm Reporting

To configure the thresholds and the type of SONET alarms that are reported, use any of the following commands. The commands listed in this section are optional. To display the current Bit Error Rate (BER) threshold setting or to view the reporting of the SONET alarms, use the **show controllers pos** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **pos threshold** {**b1-tca** | **b2-tca** | **b3-tca** | **sd-ber** | **sf-ber**} *rate*
5. **pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface pos** *slot/sub-slot/port* **Example:** `Router(config)# interface pos 2/0/0`	Specifies the POS interface to be configured as the working interface and enters interface configuration mode.
Step 4	**pos threshold** {**b1-tca** | **b2-tca** | **b3-tca** | **sd-ber** | **sf-ber**} *rate* **Example:** `Router(config-if)# pos threshold b1-tca 4`	(Optional) Configures the BER threshold values for signal failure (SF), signal degrade (SD), or threshold crossing alarms (TCAs).
Step 5	**pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**} **Example:** `Router(config-if)# pos report b2-tca`	(Optional) Enables reporting of selected SONET alarms.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring LAIS as an APS Switchover Trigger

When you place the working interface into administrative shutdown state, the switchover happens with or without **pos ais-shut**. When **pos ais-shut** is enabled on the interface, the interface sends the line alarm indication signal (LAIS) alarm to the remote end of the administrative shutdown, and the LAIS alarm makes the switchover bit faster. The **carrier-delay msec** *milliseconds* command and **ppp timeout retry seconds** [*milliseconds*] command are also used to make the APS switchover happen faster.

The **carrier-delay msec** *milliseconds* command delays the link down event processing for POS interfaces. For example, if the carrier delay is set to 50 milliseconds (ms), the router will ignore all link down events that are cleared within 50 msec. If the link goes down there will be no APS switchover for 50 ms. The default carrier delay is 2 seconds and there will be no APS switchover for 2 seconds after the link goes down. Hence the carrier delay is set to 50 ms for faster switchover.

The **ppp timeout retry seconds** [*milliseconds*] command sets the PPP retry timeout to the specified time. For example, if the timeout retry is set to 200 ms, the router tries to establish PPP link in 200 ms after it detects the signal outage due to APS switchover. If the default retry timeout of 2 seconds is used, then the PPP link will be established 2 seconds after the APS switchover. Hence the PPP timeout retry is set to 50 ms for faster switchover.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **pos ais-shut**
5. **carrier-delay msec** *milliseconds*
6. **ppp timeout retry seconds** [*milliseconds*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface pos** *slot/sub-slot/port* **Example:** `Router(config)# interface pos 2/0/0`	Specifies the POS interface to be configured as the working interface and enters interface configuration mode.
Step 4	**pos ais-shut** **Example:** `Router(config-if)# pos ais-shut`	Sends line alarm indication signal (LAIS) alarm on Admin shut of the interface.
Step 5	**carrier-delay msec** *milliseconds* **Example:** `Router(config-if)# carrier-delay msec 50`	Delays the link down event processing for POS interfaces and makes the APS switchover faster.
Step 6	**ppp timeout retry seconds** [*milliseconds*] **Example:** `Router(config-if)# ppp timeout retry 0 200`	Sets the maximum waiting period for a response during PPP negotiation and makes the APS switchover faster.

	Command or Action	Purpose
Step 7	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for 1+1 SR-APS Without Bridging

Example Configuring 1+1 SR-APS Without Bridging

The following example shows the configuration sequence for 1+1 SR-APS:

```
interface loopback 1
ip address 1.1.1.1 255.255.255.0
interface pos 2/0/0
     aps group 1
     aps working 1
     pos ais-shut
     end
interface pos 3/0/0
     aps group 1
     aps protect 1 1.1.1.1
     pos ais-shut
     end
```

The following example shows the sample output of APS configured on a router with a working interface:

```
Router# show aps
POS2/1/1 APS Group 0: protect channel 0 (Inactive)
        Working channel 1 at 10.0.1.1 (Enabled)
        bidirectional, revertive (60 seconds)
        PGP timers (default): hello time=1; hold time=3
                hello fail revert time=120
        SONET framing; SONET APS signalling by default
        Received K1K2: 0x00 0x05
                No Request (Null)
        Transmitted K1K2: 0x00 0x05
                No Request (Null)
        Remote APS configuration: (null)
POS2/1/0 APS Group 0: working channel 1 (Active)
        Protect at 10.0.1.1
        PGP timers (from protect): hello time=1; hold time=3
        SONET framing
        Remote APS configuration: (null)
```

The following example shows the display of POS controllers:

```
Router# show controller pos 2/1/0
POS2/1/0
SECTION
  LOF = 0          LOS    = 1                           BIP(B1) = 0
LINE
  AIS = 2          RDI    = 2          FEBE = 14        BIP(B2) = 0
PATH
```

```
  AIS = 2          RDI    = 2          FEBE = 4          BIP(B3) = 6
  PLM = 0          UNEQ   = 0          TIM  = 0          TIU     = 0
  LOP = 1          NEWPTR = 2          PSE  = 0          NSE     = 0
Active Defects: None
Active Alarms:  None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA
Framing: SONET
APS
 working (active)
  COAPS = 13         PSBF = 0
  State: PSBF_state = False
  Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
  Rx Synchronization Status S1 = 00
  S1S0 = 00, C2 = CF
  Remote aps status (none); Reflected local aps status (none)
CLOCK RECOVERY
  RDOOL = 0
  State: RDOOL_state = False
PATH TRACE BUFFER: STABLE
  Remote hostname : SPA-APS2
  Remote interface: POS2/2/0
  Remote IP addr  : 10.1.1.1
  Remote Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
BER thresholds:  SF = 10e-3  SD = 10e-6
TCA thresholds:  B1 = 10e-6  B2 = 10e-6  B3 = 10e-6
Clock source:  internal
```

The following example shows the configuration information and statistics for a POS interface:

```
Router# show interface pos 2/1/0
POS2/1/0 is up, line protocol is up  (APS working - active)
  Hardware is SPA-4XOC12-POS
  Internet address is 10.1.1.2/24
  MTU 4470 bytes, BW 155000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Scramble disabled
  Last input 00:00:02, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     102477 packets input, 2459448 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 4 giants, 0 throttles 0 parity
     4 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     102486 packets output, 2459934 bytes, 0 underruns
     0 output errors, 0 applique, 2 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     10 carrier transitions
```

Additional References

Related Documents

Related Topic	Document Title
APS commands	*Cisco IOS Interface and Hardware Component Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for 1+1 SR-APS Without Bridging

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 317: Feature Information for 1+1 SR-APS Without Bridging

Feature Name	Releases	Feature Information
1+1 SR-APS Without Bridging	Cisco IOS XE Release 3.1S	This feature provides support to 1+1 single router APS without bridging. There were no commands introduced or modified by this feature.

CHAPTER 262

IPv6 Rapid Deployment

The IPv6 rapid deployment feature allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4.

Information About IPv6 Rapid Deployment

IPv6 Rapid Deployment Tunnels

The IPv6 Rapid Deployment (6RD) feature is an extension of the 6to4 feature. The 6RD feature allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4.

The main differences between 6RD and 6to4 tunneling are as follows:

- 6RD does not require addresses to have a 2002::/16 prefix; therefore, the prefix can be from theservice provider's own address block. This function allows the 6RD operational domain to be within the SP network. From the perspective of customer sites and the general IPv6 Internet connected to a 6RD-enabled service provider network, the IPv6 service provided is equivalent to the native IPv6.
- All 32 bits of the IPv4 destination need not be carried in the IPv6 payload header. The IPv4 destination is obtained from a combination of bits in the payload header and information on the router. Furthermore, the IPv4 address is not at a fixed location in the IPv6 header as it is in 6to4.

How to Configure IPv6 Rapid Deployment

Configuring 6RD Tunnels

Perform this task to configure 6RD tunnels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
5. **tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**]
6. **tunnel 6rd prefix** *ipv6-prefix* / *prefix-length*
7. **tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 1`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**tunnel source** {*ip-address*| *interface-t ype interface-number*} **Example:** `Router(config-if)# tunnel source Ethernet2/0`	Specifies the source interface type and number for the tunnel interface.
Step 5	**tunnel mode ipv6ip** [**6rd**	**6to4**	**auto-tunnel**	**isatap**] **Example:** `Router(config-if)# tunnel mode ipv6ip 6rd`	Configures a static IPv6 tunnel interface.
Step 6	**tunnel 6rd prefix** *ipv6-prefix* / *prefix-length* **Example:** `Router(config-if)# tunnel 6rd prefix 2001:B000::/32`	Specifies the common IPv6 prefix on IPv6 rapid 6RD tunnels.
Step 7	**tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length* **Example:**	Specifies the prefix length and suffix length of the IPv4 transport address common to all the 6RD routers in a domain.

	Command or Action	Purpose
	`Router(config-if)# tunnel 6rd ipv4 prefix-length 16 suffix 8`	

Configuration Examples for IPv6 Rapid Deployment

Example: Configuring 6RD Tunnels

The following example shows the running configuration of a 6RD tunnel and the corresponding output of the **show tunnel 6rd** command:

```
interface Tunnel1
 ipv6 address 2001:B000:100::1/32
 tunnel source Ethernet2/1
 tunnel mode ipv6ip 6rd
 tunnel 6rd prefix 2001:B000::/32
 tunnel 6rd ipv4 prefix-len 16 suffix-len 8
end
Router# show tunnel 6rd tunnel 1
Interface Tunnel1:
  Tunnel Source: 10.1.1.1
  6RD: Operational, V6 Prefix: 2001:B000::/32
       V4 Common Prefix Length: 16, Value: 10.1.0.0
       V4 Common Suffix Length: 8, Value: 0.0.0.1
```

Feature Information for IPv6 Rapid Deployment

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 318: Feature Information for IPv6 Rapid Deployment

Feature Name	Releases	Feature Information
IP Tunneling: 6RD IPv6 Rapid Deployment	15.1(3)T	The 6RD feature allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4. The following commands were introduced or modified: **tunnel 6rd ipv4**, **tunnel 6rd prefix**, **tunnel mode ipv6ip**, **tunnel source**.

CHAPTER 263

IPv6 Automatic 6to4 Tunnels

This feature provides support for IPv6 automatic 6to4 tunnels. An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks.

Information About IPv6 Automatic 6to4 Tunnels

Automatic 6to4 Tunnels

An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks. The key difference between automatic 6to4 tunnels and manually configured tunnels is that the tunnel is not point-to-point; it is point-to-multipoint. In automatic 6to4 tunnels, routers are not configured in pairs because they treat the IPv4 infrastructure as a virtual nonbroadcast multiaccess (NBMA) link. The IPv4 address embedded in the IPv6 address is used to find the other end of the automatic tunnel.

An automatic 6to4 tunnel may be configured on a border router in an isolated IPv6 network, which creates a tunnel on a per-packet basis to a border router in another IPv6 network over an IPv4 infrastructure. The tunnel destination is determined by the IPv4 address of the border router extracted from the IPv6 address that starts with the prefix 2002::/16, where the format is 2002:*border-router-IPv4-address* ::/48. Following the embedded IPv4 address are 16 bits that can be used to number networks within the site. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks. 6to4 tunnels are configured between border routers or between a border router and a host.

The simplest deployment scenario for 6to4 tunnels is to interconnect multiple IPv6 sites, each of which has at least one connection to a shared IPv4 network. This IPv4 network could be the global Internet or a corporate backbone. The key requirement is that each site have a globally unique IPv4 address; the Cisco software uses this address to construct a globally unique 6to4/48 IPv6 prefix. As with other tunnel mechanisms, appropriate entries in a Domain Name System (DNS) that map between hostnames and IP addresses for both IPv4 and IPv6 allow the applications to choose the required address.

How to Configure IPv6 Automatic 6to4 Tunnels

Configuring Automatic 6to4 Tunnels

Before you begin

With 6to4 tunnels, the tunnel destination is determined by the border router IPv4 address, which is concatenated to the prefix 2002::/16 in the format 2002:*border-router-IPv4-address* ::/48. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks.

Note The configuration of only one IPv4-compatible tunnel and one 6to4 IPv6 tunnel is supported on a router. If you choose to configure both of those tunnel types on the same router, we strongly recommend that they do not share the same tunnel source.

The reason that a 6to4 tunnel and an IPv4-compatible tunnel cannot share an interface is that both of them are NBMA "point-to-multipoint" access links and only the tunnel source can be used to reorder the packets from a multiplexed packet stream into a single packet stream for an incoming interface. So when a packet with an IPv4 protocol type of 41 arrives on an interface, that packet is mapped to an IPv6 tunnel interface based on the IPv4 address. However, if both the 6to4 tunnel and the IPv4-compatible tunnel share the same source interface, the router is not able to determine the IPv6 tunnel interface to which it should assign the incoming packet.

IPv6 manually configured tunnels can share the same source interface because a manual tunnel is a "point-to-point" link, and both the IPv4 source and IPv4 destination of the tunnel are defined.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** {*ipv6-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length*
5. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
6. **tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**
7. **exit**
8. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix* / *prefix-length*{*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 1`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits*/*prefix-length* **Example:** `Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
Step 5	**tunnel source** {*ip-address*| *interface-t ype interface-number*} **Example:** `Router(config-if)# tunnel source loopback 1`	Specifies the source interface type and number for the tunnel interface.
Step 6	**tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap** **Example:** `Router(config-if)# tunnel mode ipv6ip 6rd`	Configures a static IPv6 tunnel interface. • The **auto-tunnel** keyword is not supported on Cisco ASR 1000 series routers.
Step 7	**exit** **Example:** `Router(config-if) exit`	Exits interface configuration mode, and enters global configuration mode.
Step 8	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix / prefix-length*{*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] **Example:** `Router(config)# ipv6 route 2002::/16 tunnel 0`	Configures a static route for the IPv6 6to4 prefix 2002::/16 to the specified tunnel interface. **Note** When configuring a 6to4 overlay tunnel, you must configure a static route for the IPv6 6to4 prefix 2002::/16 to the 6to4 tunnel interface. • The tunnel number specified in the **ipv6 route** command must be the same tunnel number specified in the **interface tunnel** command.

Configuration Examples for IPv6 Automatic 6to4 Tunnels

Example: Configuring 6to4 Tunnels

The following example configures a 6to4 tunnel on a border router in an isolated IPv6 network. The IPv4 address is 192.168.99.1, which translates to the IPv6 prefix of 2002:c0a8:6301::/48. The IPv6 prefix is subnetted into 2002:c0a8:6301::/64 for the tunnel interface: 2002:c0a8:6301:1::/64 for the first IPv6 network, and 2002:c0a8:6301:2::/64 for the second IPv6 network. The static route ensures that any other traffic for the IPv6 prefix 2002::/16 is directed to tunnel interface 0 for automatic tunneling.

```
interface GigabitEthernet0/0/0
 description IPv4 uplink
 ip address 192.168.99.1 255.255.255.0
!
interface GigabitEthernet1/0/0
 description IPv6 local network 1
 ipv6 address 2002:c0a8:6301:1::1/64
!
interface GigabitEthernet2/0/0
 description IPv6 local network 2
 ipv6 address 2002:c0a8:6301:2::1/64
!
interface Tunnel0
 description IPv6 uplink
 no ip address
 ipv6 address 2002:c0a8:6301::1/64
 tunnel source GigabitEthernet0/0/0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 tunnel 0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: *http://www.cisco.com/go/mibs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Automatic 6to4 Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 319: Feature Information for IPv6 Automatic 6to4 Tunnels

Feature Name	Releases	Feature Information
IPv6 Tunneling: Automatic 6to4 Tunnels	Cisco IOS XE Release 2.1	An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks. The following commands were introduced or modified: **tunnel mode ipv6ip**, **tunnel source**.

CHAPTER 264

GRE IPv6 Tunnels

The GRE IPv6 Tunnels feature enables the delivery of packets from other protocols through an IPv6 network and allows the routing of IPv6 packets between private networks across public networks with globally routed IPv6 addresses. Generic routing encapsulation (GRE) is a unicast protocol that offers the advantages of encapsulating broadcast and multicast traffic (multicast streaming or routing protocols) or other non-IP protocols and of being protected by IPsec.

Restrictions for GRE IPv6 Tunnels

- GRE tunnel keepalive packets are not supported.
- Multipoint GRE (mGRE) IPv6 tunneling is not supported.

Information About GRE IPv6 Tunnels

Overview of GRE IPv6 Tunnels

The GRE IPv6 Tunnels feature enables the delivery of packets from other protocols through an IPv6 network and allows the routing of IPv6 packets between private networks across public networks with globally routed IPv6 addresses.

For point-to-point GRE tunnels, each tunnel interface requires a tunnel source IPv6 address and a tunnel destination IPv6 address when being configured. All packets are encapsulated with an outer IPv6 header and a GRE header.

GRE IPv6 Tunnel Protection

GRE IPv6 tunnel protection allows devices to work as security gateways, establish IPsec tunnels between other security gateway devices, and provide crypto IPsec protection for traffic from internal networks when the traffic is sent across the public IPv6 Internet. The GRE IPv6 tunnel protection functionality is similar to the security gateway model that uses GRE IPv4 tunnel protection.

How to Configure GRE IPv6 Tunnels

Configure CDP Over GRE IPv6 Tunnels

Perform this task to configure a GRE tunnel on an IPv6 network. GRE tunnels can be configured to run over an IPv6 network layer and transport IPv6 and IPv4 packets through IPv6 tunnels.

Note You must enable IPv6 or configure IPv6 MTU size more than 1500 on a tunnel's exit interface to avoid receiving warning messages.

Before you begin

When GRE IPv6 tunnels are configured, IPv6 addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 or IPv6 addresses. The host or device at each end of the configured tunnel must support both IPv4 and IPv6 protocol stacks.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**CDP enable** **Example:** `Device(config)# CDP enable`	Enables Cisco Discovery Protocol on the interface.
Step 5	**tunnel source** {*ipv6-address* | *interface-type* |*interface-number* }	Specifies the source IPv6 address or the source interface type and number for the tunnel interface.

	Command or Action	Purpose
	Example: `Device(config-if)# tunnel source ethernet 0`	• If an interface type and number are specified, the interface must be configured with an IPv6 address. **Note** For more information on the tunnel source command, refer to the IPv6 command reference guide.
Step 6	**tunnel destination** *ipv6-address* **Example:** `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	Specifies the destination IPv6 address for the tunnel interface. **Note** For more information on the tunnel destination command, refer to the IPv6 command reference guide.
Step 7	**tunnel mode gre ipv6** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel interface. Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring GRE IPv6 Tunnel Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ipv6-address* | *interface-type interface-number*}
5. **tunnel destination** *ipv6-address*
6. **tunnel mode gre ipv6**
7. **tunnel protection ipsec profile** *profile-name*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**tunnel source** {*ipv6-address* | *interface-type interface-number*} **Example:** `Device(config-if)# tunnel source ethernet 0`	Specifies the source IPv6 address or the source interface type and number for the tunnel interface. • If an interface type and number are specified, the interface must be configured with an IPv6 address. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 5	**tunnel destination** *ipv6-address* **Example:** `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	Specifies the destination IPv6 address for the tunnel interface. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 6	**tunnel mode gre ipv6** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel interface. Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 7	**tunnel protection ipsec profile** *profile-name* **Example:** `Device(config-if)# tunnel protection ipsec profile ipsec-profile`	Associates the tunnel interface with an IPsec profile. **Note** For the *profile-name* argument, specify the IPsec profile configured in global configuration mode.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for GRE IPv6 Tunnels

Example: Configuring CDP Over GRE IPv6 Tunnels

The following example shows how to configure a GRE tunnel over an IPv6 transport. In this example, Ethernet0/0 has an IPv6 address, and this is the source address used by the tunnel interface. The destination IPv6 address of the tunnel is specified directly. In this example, the tunnel carries both IPv4 and IS-IS traffic.

```
interface Tunnel0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
 tunnel source Ethernet0/0
 tunnel destination 2001:DB8:1111:2222::1
 tunnel mode gre ipv6
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:1111:1111::1/64
!
router isis
 net 49.0001.0000.0000.000a.00
```

The following example shows how to configure CDP on GRE IPv6 P2P Tunnel Interface.

```
interface Tunnel1
  cdp enable
  ipv6 address 20::1/64
  tunnel source Ethernet0/0
  tunnel mode gre ipv6
  tunnel destination 10::2
end
```

The following example shows how to configure CDP on GRE IPv6 Multipoint Tunnel Interface.

```
interface Tunnel1
   ipv6 address 172::2/64
   ipv6 nhrp map 172::1/64 192::1
   ipv6 nhrp map multicast 192::1
    ipv6 nhrp network-id 1
   ipv6 nhrp nhs 172::1
   llp nhrp map multicast 192::1
   tunnel source 2000::1
   tunnel mode gre multipoint ipv6
end
```

The following show example displays the CDP neighbor tunnels that are configured in a device.

```
Router#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Router           Tunnel1           179               R    Linux Uni Tunnel1
```

Example: Configuring GRE IPv6 Tunnel Protection

The following example shows how to associate the IPsec profile “ipsec-profile” with a GRE IPv6 tunnel interface. The IPsec profile is configured using the **crypto ipsec profile** command.

```
crypto ipsec profile ipsec-profile
 set transform-set ipsec-profile
!
interface Tunnel1
 ip address 192.168.1.1 255.255.255.252
 tunnel source FastEthernet2/0
 tunnel destination 10.13.7.67
 tunnel protection ipsec profile ipsec-profile
```

Information About EoMPLS over IPv6 GRE Tunnel

Ethernet over MPLS (EoMPLS) is a tunneling mechanism that allows you to tunnel Layer 2 traffic through a Layer 3 MPLS network. EoMPLS is also known as Layer 2 tunneling.

The EoMPLS over IPv6 GRE Tunnel feature supports tunneling of EoMPLS traffic via an IPv6 network by using GRE tunnels. Effective from Cisco IOS XE Release 3.15s, EoMPLS is supported over IPv6 GRE tunnel.

Configuring EoMPLS over IPv6 GRE Tunnel

EoMPLS over IPv6 GRE Tunnel can be configured in the following two methods:

Using Legacy Commands, on page 3156

Using Protocol-based Commands, on page 3158

Using Legacy Commands

This section describes how to configure EoMPLS over IPv6 GRE Tunnel using legacy commands. The following are relevant configurations from both Provider Edge 1 Router and Provider Edge 2 Router:

SUMMARY STEPS

1. configure terminal
2. ipv6 unicast-routing
3. mpls label protocol ldp
4. mpls ldp router-id Loopback0 [force]
5. interface *type number*
6. ip address *ip-address mask*
7. interface gigabitethernet slot/port
8. encapsulation dot1 *vlan-id*
9. xconnect *peer-ipaddress vc-id* encapsulation mpls
10. interface tunnel *interface number*
11. ip address *ip-address mask*
12. tunnel source {*ip-address* | *interface-type interface-number*}
13. tunnel mode gre ipv6

14. tunnel destination *ipv6-address*
15. mpls ip
16. interface gigabitethernet slot/port
17. ipv6 address { *ipv6-prefix*/*prefix-length* | *prefix-name sub-bits*/*prefix-length* }

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Router#configure terminal`	Enters global configuration mode.
Step 2	ipv6 unicast-routing **Example:** `Router(config)#ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams globally on the router.
Step 3	mpls label protocol ldp **Example:** `Router(config)#mpls label protocol ldp`	Enables Label Distribution Protocol (LDP).
Step 4	mpls ldp router-id Loopback0 [force] **Example:** `Router(config)#mpls ldp router-id Loopback0 [force]`	Configures the LDP Router ID. **Note** The optional force keyword ensures that the IP address on interface loopback 0, and not the IP address of any other interface, becomes the LDP router ID.
Step 5	interface *type number* **Example:** `Router(config)#interface Loopback 0`	Enters configuration mode for the loopback interface.
Step 6	ip address *ip-address mask* **Example:** `Router(config-if)#ip address 10.1.1.2 255.255.255.255`	Sets the IP address and subnet mask for the loopback interface.
Step 7	interface gigabitethernet slot/port **Example:** `Router(config-if)#interface GigabitEthernet0/0/1.2`	Enters the configuration mode for a Gigabit Ethernet interface on the router.
Step 8	encapsulation dot1 *vlan-id* **Example:** `Router(config-subif)#encapsulation dot1q 200`	Enables 802.1Q trunking on a router.
Step 9	xconnect *peer-ipaddress vc-id* encapsulation mpls **Example:**	Enables the attachment circuit and specifies the IP address of the peer, a VC ID, and the data encapsulation method.

	Command or Action	Purpose
	`Router(config-subif)#xconnect 10.1.1.1 100 encapsulation mpls`	
Step 10	interface tunnel *interface number* **Example:** `Router(config)#interface tunnel 10`	Designates a tunnel interface and enters interface configuration mode.
Step 11	ip address *ip-address mask* **Example:** `Router(config-if)#ip address 192.0.2.1 255.255.255.0`	Sets the IP address and subnet mask for the loopback interface.
Step 12	tunnel source {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)#tunnel source GigabitEthernet 0/0/0`	Specifies the source IPv4 address or the source interface type and number for the tunnel interface.
Step 13	tunnel mode gre ipv6 **Example:** `Router (config-if)#tunnel mode gre ipv6`	Specifies that the GRE over IPv6 encapsulation protocol is used in the tunnel.
Step 14	tunnel destination *ipv6-address* **Example:** `Router(config-if)#tunnel destination 2002::2`	Specifies the destination IPv6 address for the tunnel interface.
Step 15	mpls ip **Example:** `Router(config-if)#mpls ip`	Enables mpls processing on the tunnel interface.
Step 16	interface gigabitethernet slot/port **Example:** `Router(config-if)#interface GigabitEthernet0/0/0`	Enters the configuration mode for a Gigabit Ethernet interface on the router.
Step 17	ipv6 address { *ipv6-prefix*/*prefix-length* | *prefix-name sub-bits*/*prefix-length* } **Example:** `Router(config-if)#ipv6 address 2002::1/112`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Example

Using Protocol-based Commands

This section describes how to configure EoMPLS over IPv6 GRE Tunnel using Protocol-based commands.

SUMMARY STEPS

1. template type pseudowire [*pseudowire-name*]
2. encapsulation mpls
3. end
4. interface pseudowire *number*
5. source template type pseudowire
6. encapsulation mpls
7. neighbor *peer-address vcid-value*
8. end
9. l2vpn xconnect context *context-name*
10. member pseudowire *interface-number*
11. member gigabit ethernet *interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	template type pseudowire [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire eompls`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 2	encapsulation mpls **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 3	end **Example:** `Router(config-pw-class)# end`	Exits to privileged EXEC mode.
Step 4	interface pseudowire *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 5	source template type pseudowire **Example:** `Router(config-if)# source template type pseudowire eompls`	Configures the source template of type pseudowire named EoMPLS.
Step 6	encapsulation mpls **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 7	neighbor *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 154.154.154.154 100`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 8	end	Exits to privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router(config-if)# end`	
Step 9	l2vpn xconnect context *context-name* **Example:** `Router(config)# l2vpn xconnect context eompls_100`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 10	member pseudowire *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 11	member gigabit ethernet *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet0/0/1`	Specifies the location of the Gigabit Ethernet member interface.

Example

Verifying the EoMPLS over IPv6 GRE Tunnel Configuration

Use the following commands to verify that the EoMPLS over IPv6 GRE Tunnel feature is correctly configured.

SUMMARY STEPS

1. show inter tunnel [*tunnel-id*]
2. show xconnect all [detail]
3. show mpls l2transport vc id detail

DETAILED STEPS

	Command or Action	Purpose

Step 1	show inter tunnel [*tunnel-id*]	`Router# show inter tunnel10`
		`Tunnel10 is up, line protocol is up`
		`  Hardware is Tunnel`
		`  Internet address is 192.0.2.1/24`
		`  MTU 1456 bytes, BW 100 Kbit/sec, DLY 50000 usec,`
		
		`     reliability 255/255, txload 1/255, rxload 1/255`
		`  Encapsulation TUNNEL, loopback not set`
		`  Keepalive not set`
		`  Tunnel linestate evaluation up`
		`  Tunnel source 2002::2 (GigabitEthernet0/0/0), destination 2002::1`
		`   Tunnel Subblocks:`
		`      src-track:`
		`         Tunnel10 source tracking subblock associated with GigabitEthernet0/0/0`
		`          Set of tunnels with source`

	Command or Action	Purpose

		GigabitEthernet0/0/0, 1 member (includes iterators), on interface <OK>
		  Tunnel protocol/transport GRE/IPv6
		  Key disabled, sequencing disabled
		  Checksumming of packets disabled
		  Tunnel TTL 255
		  Path MTU Discovery, ager 10 mins, min MTU 1280
		  Tunnel transport MTU 1456 bytes
		  Tunnel transmit bandwidth 8000 (kbps)
		  Tunnel receive bandwidth 8000 (kbps)
		  Last input never, output never, output hang never
		
		  Last clearing of "show interface" counters 04:41:12
		  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
		  Queueing strategy: fifo
		  Output queue: 0/0 (size/max)
		  30 second input rate 0 bits/sec, 0 packets/sec
		  30 second output rate 0 bits/sec, 0 packets/sec
		  8363 packets input, 1074130 bytes, 0 no buffer
		
		  Received 0 broadcasts (0 IP multicasts)
		  0 runts, 0 giants, 0 throttles
		  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
		  8384 packets output, 1076628 bytes, 0 underruns
		  0 output errors, 0 collisions, 0 interface resets
		  0 unknown protocol drops
		  0 output buffer failures, 0 output buffers swapped out

Step 2	show xconnect all [detail]	Router# show xconnect all
		Legend: XC ST=Xconnect State S1=Segment1 State
		  S2=Segment2 State
		  UP=Up DN=Down AD=Admin Down
		  IA=Inactive
		  SB=Standby HS=Hot Standby RV=Recovering
		  NH=No Hardware
		
		XC ST Segment 1 S1 Segment 2 S2
		——+————————+-+————————+—
		UP pri ac Gi0/0/0.2:200(Eth VLAN) UP mpls 10.1.1.2:100 UP
		
		asr1001#show xconnect all detail
		Legend: XC ST=Xconnect State S1=Segment1 State
		  S2=Segment2 State
		  UP=Up DN=Down AD=Admin Down
		  IA=Inactive
		  SB=Standby HS=Hot Standby RV=Recovering
		  NH=No Hardware
		
		XC ST Segment 1 S1 Segment 2 S2
		——+————————+-+————————+—
		UP pri ac Gi0/0/0.2:200(Eth VLAN) UP mpls 10.1.1.2:100 UP
		  Interworking: ethernet

	Command or Action	Purpose
		Local VC label 17 Remote VC label 17
Step 3	show mpls l2transport vc id detail	Router# show mpls l2transport vc 100 detail Local interface: Gi0/0/0.2 up, line protocol up, Eth VLAN 200 up Interworking type is Ethernet Destination address: 10.1.1.2, VC ID: 100, VC status: up Output interface: Tu10, imposed label stack {17} Preferred path: not configured Default path: active Next hop: point2point Create time: 05:52:23, last status change time: 05:52:07 Last label FSM state change time: 05:52:07 Signaling protocol: LDP, peer 10.1.1.2:0 up Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.2, LDP is UP Graceful restart: configured and not enabled Non stop routing: not configured and not enabled Status TLV support (local/remote) : enabled/supported LDP route watch : enabled Label/status state machine : established, LruRru Last local dataplane status rcvd: No fault Last BFD dataplane status rcvd: Not sent Last BFD peer monitor status rcvd: No fault Last local AC circuit status rcvd: No fault Last local AC circuit status sent: No fault Last local PW i/f circ status rcvd: No fault Last local LDP TLV status sent: No fault Last remote LDP TLV status rcvd: No fault Last remote LDP ADJ status rcvd: No fault MPLS VC labels: local 17, remote 17 Group ID: local 0, remote 0 MTU: local 1500, remote 1500 Remote interface description: Sequencing: receive disabled, send disabled Control Word: On (configured: autosense) SSO Descriptor: 10.1.1.2/100, local label: 17 Dataplane: SSM segment/switch IDs: 4098/4097 (used), PWID: 1 VC statistics: transit packet totals: receive 0, send 0 transit byte totals: receive 0, send 0 transit packet drops: receive 0, seq error 0, send 0

Example

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
Tunnel commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Interface and Hardware Component Command Reference
IPv6 commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	IPv6 Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GRE IPv6 Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 265

Cisco Discovery Protocol over GRE Tunnels

This document describes how Cisco Discovery Protocol (CDP) delivers packets from other protocols through a network and allows the routing of packets between private networks across public networks with globally routed addresses. CDP is supported over Generic Routing Encapsulation (GRE) Point-to-Point tunnel interface and GRE Multipoint Tunnel interface.

Generic routing encapsulation (GRE) is a unicast protocol that offers the advantages of encapsulating broadcast and multicast traffic (multicast streaming or routing protocols) or other non-IP protocols and of being protected by IPsec. CDP is a Layer 2, media-independent, and network-independent protocol. Networking applications use CDP to know about devices that are connected directly to applications.

CDP is disabled by default on the interfaces. To enable CDP, use the **cdp enable** command in interface configuration mode.

Feature Information for CDP Over GRE Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 320: Feature Information for CDP Over GRE Tunnels

Feature Name	Releases	Feature Information
CDP Over GRE Tunnels	Cisco IOS XE Release 16.12.1	The CDP over GRE Tunnels feature enables the delivery of packets from other protocols through a network and allows the routing of packets between private networks across public networks with globally routed addresses.

Overview of CDP Over GRE Tunnels

Networking applications use CDP over GRE tunnel to identify tunnel endpoints which may not be directly connected. With this enhancement, CDP can exchange exchange neighbour addressing information over the GRE tunnel.

For point-to-point GRE tunnels, each tunnel interface requires a tunnel source address and a tunnel destination address when being configured. All packets are encapsulated with an outer header and a GRE header.

CDP provides the following benefits:

- Allows systems using different network layer protocols to learn about one another.
- Facilitates management of Cisco devices by discovering them and discovering how they are configured.
- Assists with troubleshooting Type-Length-Value Fields (TLV) fields.
- Helps to learn SNMP agent addresses and sends the SNMP queries.

Note When CDP feature is enabled on GRE Multipoint Tunnel interface, an additional command (**llp nhrp map multicast**) is required for Cisco IOS XE releases 16.12.1 or later.

Configuring CDP Over GRE Tunnels

Perform this task to configure a GRE tunnel on an IPv6 network. GRE tunnels can be configured to run over an IPv6 network layer and transport IPv6 and IPv4 packets through IPv6 tunnels.

Note You must enable IPv6 or configure IPv6 MTU size more than 1500 on a tunnel's exit interface to avoid receiving warning messages.

Before you begin

When GRE tunnels are configured, IP addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 or IPv6 addresses. The host or device at each end of the configured tunnel must support both IPv4 and IPv6 protocol stacks.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**CDP enable** **Example:** `Device(config)# CDP enable`	Enables Cisco Discovery Protocol on the interface.
Step 5	**tunnel source** {*ipv6-address* | *interface-type* |*interface-number* } **Example:** `Device(config-if)# tunnel source ethernet 0`	Specifies the source IPv6 address or the source interface type and number for the tunnel interface. • If an interface type and number are specified, the interface must be configured with an IPv6 address. **Note** For more information on the tunnel source command, refer to the IPv6 command reference guide.
Step 6	**tunnel destination** *ipv6-address* **Example:** `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	Specifies the destination IPv6 address for the tunnel interface. **Note** For more information on the tunnel destination command, refer to the IPv6 command reference guide.
Step 7	**tunnel mode gre ipv6** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel interface. Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Configuring CDP Over GRE IPv6 and IPv4 Tunnels

The following example shows how to configure a GRE tunnel over an IPv6 transport. In this example, Ethernet0/0 has an IPv6 address, and this is the source address used by the tunnel interface. The destination IPv6 address of the tunnel is specified directly. In this example, the tunnel carries both IPv4 and IS-IS traffic.

```
interface Tunnel0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
 tunnel source Ethernet0/0
 tunnel destination 2001:DB8:1111:2222::1
 tunnel mode gre ipv6
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:1111:1111::1/64
!
router isis
 net 49.0001.0000.0000.000a.00
```

The following examples show how to configure CDP on GRE IPv6 and IPv4 P2P Tunnel Interfaces.

```
interface Tunnel1
  cdp enable
  ipv6 address 2001:DB8::1/32
tunnel source Ethernet0/0
tunnel mode gre ipv6
tunnel destination 2001:DB8::2:1
end
```

```
interface Tunnel300
cdp enable
ip address 192.0.2.1
 tunnel source GigabitEthernet1
 tunnel mode gre ip
tunnel destination 198.51.100.1
end
```

The following examples show how to configure CDP on GRE IPv6 and IPv4 Multipoint Tunnel Interfaces.

```
interface Tunnel1
   ipv6 address 2001:DB8::2/32
   ipv6 nhrp map 2001:DB8::1/32 192::1
   ipv6 nhrp map multicast 192::1
    ipv6 nhrp network-id 1
   ipv6 nhrp nhs 2001:DB8::1
     cdp enable
  llp nhrp map multicast 192::1
   tunnel source 2001:DB8::1
tunnel mode gre multipoint ipv6
end
```

```
interface Tunnel20
    ip address
    no ip redirects
```

```
    ip nhrp authentication cisco1
  ip nhrp network-id 20
  ip nhrp nhs 172.x.x.1 nbma 198.51.x.x multicast
    cdp enable
  llp nhrp map multicast 198.51.x.x
  tunnel source GigabitEthernet1
  tunnel mode gre multipoint
end
```

The following show example displays the CDP neighbor tunnels that are configured in a device.

```
Router#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Router           Tunnel1           179               R    Linux Uni Tunnel1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
Tunnel commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Interface and Hardware Component Command Reference
IPv6 commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	IPv6 Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 266

ISATAP Tunnel Support for IPv6

ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6

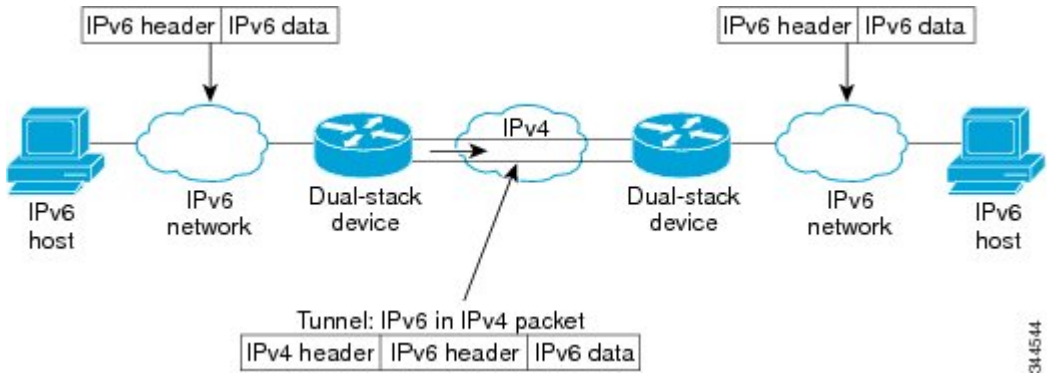
Information About ISATAP Tunnel Support for IPv6

Overlay Tunnels for IPv6

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the figure below). By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border devices or between a border device and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks. IPv6 supports the following types of overlay tunneling mechanisms:

- Manual
- Generic routing encapsulation (GRE)
- IPv4-compatible
- 6to4
- Intrasite Automatic Tunnel Addressing Protocol (ISATAP)

Figure 216: Overlay Tunnels



Note Overlay tunnels reduce the maximum transmission unit (MTU) of an interface by 20 octets (assuming that the basic IPv4 packet header does not contain optional fields). A network that uses overlay tunnels is difficult to troubleshoot. Therefore, overlay tunnels that connect isolated IPv6 networks should not be considered a final IPv6 network architecture. The use of overlay tunnels should be considered as a transition technique toward a network that supports both the IPv4 and IPv6 protocol stacks or just the IPv6 protocol stack.

Use the table below to help you determine which type of tunnel that you want to configure to carry IPv6 packets over an IPv4 network.

Table 321: Suggested Usage of Tunnel Types to Carry IPv6 Packets over an IPv4 Network

Tunneling Type	Suggested Usage	Usage Notes
Manual	Simple point-to-point tunnels that can be used within a site or between sites.	Can carry IPv6 packets only.
GRE- and IPv4-compatible	Simple point-to-point tunnels that can be used within a site or between sites.	Can carry IPv6, Connectionless Network Service (CLNS), and many other types of packets.
IPv4- compatible	Point-to-multipoint tunnels.	Uses the ::/96 prefix. We do not recommend using this tunnel type.
6to4	Point-to-multipoint tunnels that can be used to connect isolated IPv6 sites.	Sites use addresses from the 2002::/16 prefix.
6RD	IPv6 service is provided to customers over an IPv4 network by using encapsulation of IPv6 in IPv4.	Prefixes can be from the SP's own address block.
ISATAP	Point-to-multipoint tunnels that can be used to connect systems within a site.	Sites can use any IPv6 unicast addresses.

Individual tunnel types are discussed in detail in this document. We recommend that you review and understand the information about the specific tunnel type that you want to implement. When you are familiar with the

type of tunnel you need, see the table below for a summary of the tunnel configuration parameters that you may find useful.

Table 322: Tunnel Configuration Parameters by Tunneling Type

Tunneling Type	Tunnel Configuration Parameter			
Tunnel Mode	**Tunnel Source**	**Tunnel Destination**	**Interface Prefix or Address**	
Manual	ipv6ip	An IPv4 address, or a reference to an interface on which IPv4 is configured.	An IPv4 address.	An IPv6 address.
GRE/IPv4	gre ip		An IPv4 address.	An IPv6 address.
IPv4- compatible	ipv6ip auto-tunnel		Not required. These are all point-to-multipoint tunneling types. The IPv4 destination address is calculated, on a per-packet basis, from the IPv6 destination.	Not required. The interface address is generated as ::*tunnel-source*/96.
6to4	ipv6ip 6to4			An IPv6 address. The prefix must embed the tunnel source IPv4 address.
6RD	ipv6ip 6rd			An IPv6 address.
ISATAP	ipv6ip isatap			An IPv6 prefix in modified eui-64 format. The IPv6 address is generated from the prefix and the tunnel source IPv4 address.

ISATAP Tunnels

ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6. ISATAP is designed for transporting IPv6 packets *within* a site where a native IPv6 infrastructure is not yet available; for example, when sparse IPv6 hosts are deployed for testing. ISATAP tunnels allow individual IPv4 or IPv6 dual-stack hosts within a site to communicate with other such hosts on the same virtual link, basically creating an IPv6 network using the IPv4 infrastructure.

The ISATAP router provides standard router advertisement network configuration support for the ISATAP site. This feature allows clients to automatically configure themselves as they would do if they were connected to a GigabitEthernet or FastEthernet. It can also be configured to provide connectivity out of the site. ISATAP uses a well-defined IPv6 address format composed of any unicast IPv6 prefix (/64), which can be link local, or global (including 6to4 prefixes), enabling IPv6 routing locally or on the Internet. The IPv4 address is encoded in the last 32 bits of the IPv6 address, enabling automatic IPv6-in-IPv4 tunneling.

Although the ISATAP tunneling mechanism is similar to other automatic tunneling mechanisms, such as IPv6 6to4 tunneling, ISATAP is designed for transporting IPv6 packets *within* a site, not *between* sites.

ISATAP uses unicast addresses that include a 64-bit IPv6 prefix and a 64-bit interface identifier. The interface identifier is created in modified EUI-64 format in which the first 32 bits contain the value 000:5EFE to indicate that the address is an IPv6 ISATAP address. The table below describes an ISATAP address format.

Table 323: IPv6 ISATAP Address Format

64 Bits	32 Bits	32 Bits
link local or global IPv6 unicast prefix	0000:5EFE	IPv4 address of the ISATAP link

As shown in the table above, an ISATAP address consists of an IPv6 prefix and the ISATAP interface identifier. This interface identifier includes the IPv4 address of the underlying IPv4 link. The following example shows what an actual ISATAP address would look like if the prefix is 2001:DB8:1234:5678::/64 and the embedded IPv4 address is 10.173.129.8. In the ISATAP address, the IPv4 address is expressed in hexadecimal as 0AAD:8108:

2001:DB8:1234:5678:0000:5EFE:0AAD:8108

How to Configure ISATAP Tunnel Support for IPv6

Configuring ISATAP Tunnels

Before you begin

The **tunnel source** command used in the configuration of an ISATAP tunnel must point to an interface with an IPv4 address configured. The ISATAP IPv6 address and prefix (or prefixes) advertised are configured as for a native IPv6 interface. The IPv6 tunnel interface must be configured with a modified EUI-64 address because the last 32 bits in the interface identifier are constructed using the IPv4 tunnel source address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits/prefix-length*
5. **no ipv6 nd ra suppress**
6. **tunnel source** {*ip-address*| *interface-type interface-number*}
7. **tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 1`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**ipv6 address** {*ipv6-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length* **Example:** `Router(config-if)# ipv6 address 2001:DB8:6301::/64 eui-64`	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.
Step 5	**no ipv6 nd ra suppress** **Example:** `Router(config-if)# no ipv6 nd ra suppress`	Sending of IPv6 router advertisements is disabled by default on tunnel interfaces. This command reenables the sending of IPv6 router advertisements to allow client autoconfiguration.
Step 6	**tunnel source** {*ip-address*| *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source gigabitethernet 1/0/1`	Specifies the source interface type and number for the tunnel interface. **Note** The interface type and number specified in the **tunnel source** command must be configured with an IPv4 address.
Step 7	**tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap** **Example:** `Router(config-if)# tunnel mode ipv6ip isatap`	Specifies an IPv6 overlay tunnel using a ISATAP address. • The **auto-tunnel** keyword is not supported on Cisco ASR 1000 series routers.

Configuration Examples for ISATAP Tunnel Support for IPv6

Example: Configuring ISATAP Tunnels

The following example shows the tunnel source defined on GigabitEthernet 0/0/0 and the **tunnel mode** command used to configure the ISATAP tunnel. Router advertisements are enabled to allow client autoconfiguration.

```
ipv6 unicast-routing
interface tunnel 1
 tunnel source Gigabitethernet 0/0/0
 tunnel mode ipv6ip isatap
 ipv6 address 2001:DB8::/64 eui-64
 no ipv6 nd ra suppress
 exit
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*Cisco IOS IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Standards and RFCs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for ISATAP Tunnel Support for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 324: Feature Information for ISATAP Tunnel Support for IPv6

Feature Name	Releases	Feature Information
ISATAP Tunnel Support for IPv6	Cisco IOS XE Release 2.1	ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6. The following commands were introduced or modified: **ipv6 nd ra suppress**, **tunnel mode ipv6ip**, **tunnel source**.

CHAPTER 267

VRF-Aware Tunnels

Virtual Routing and Forwarding (VRF)-aware tunnels are used to connect customer networks separated by untrusted core networks or core networks with different infrastructures (IPv4 or IPv6).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for VRF-Aware Tunnels

- You must configure customer edge networks. See the Configuring Customer Edge Networks for Tunneling , on page 3185 section.
- You must configure the customer and transport VRFs. See the Defining a VRF Instance, on page 3184 section.

Information About VRF-Aware Tunnels

Tunnel IP Source and Destination VRF Membership

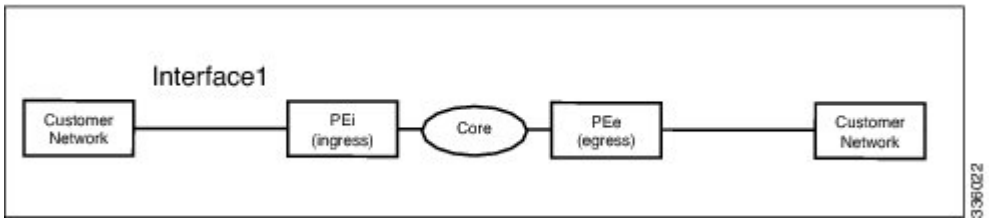
You can configure the source and destination of a tunnel to belong to any VPN routing and forwarding (VRFs) tables. A VRF table stores routing data for each VPN. The VRF table defines the VPN membership of a customer site that is attached to the network access server (NAS). Each VRF table comprises an IP routing table, a derived Cisco Express Forwarding table, and guidelines and routing protocol parameters that control the information that is included in the routing table.

You can configure the tunnel source and destination to belong to any VRF or to a global table. The tunnel becomes disabled if no route to the tunnel destination is defined.

VRF-Aware Tunnels

Virtual Routing and Forwarding (VRF)-aware tunnels are used to connect customer networks that are separated by untrusted IPv4 or IPv6 core networks.

Figure 217: VRF-Aware Tunnels



In the above topology, a tunnel is configured in the core network. Provider edge (PE) device PEi, is the tunnel head for packets entering on Interface 1. PE device PEe, is the tunnel tail for packets entering on Interface 1.

The VRF configured on Interface 1 is the customer VRF. Packets entering through Interface 1 are routed using this VRF. Packets exiting the tunnel are forwarded to this VRF. The routing by the customer VRF is called inner IP packet routing.

The VRF configured on the tunnel using the **tunnel vrf** command is the transport VRF. The transport VRF is the VRF that applies to the encapsulated payload and is used to look up the tunnel endpoints. This VRF is the same as the VRF associated with the physical interface over which the tunnel sends packets. The routing by the transport VRF is the outer IP packet routing.

The tunnel endpoint can be configured as an address from the global routing table or an address from a configured transport VRF table.

VRF-Aware IPv6 over IPv6 Tunnels

You can forward IPv6 packets on an untrusted IPv6 infrastructure by creating Virtual Routing and Forwarding (VRF)-aware IPv6 tunnels in it. These tunnels can have endpoints in a VRF table or in a global routing table. The tunnel modes used are **tunnel mode gre ipv6** and **tunnel mode ipv6**.

VRF-Aware IPv4 over IPv6 Tunnels

You can forward IPv4 packets on an untrusted IPv6 infrastructure by creating Virtual Routing and Forwarding (VRF)-aware IPv4 tunnels in it. These tunnels can have endpoints in a VRF table or in a global routing table. The tunnel modes used are **tunnel mode gre ipv6** and **tunnel mode ipv6**.

VRF-Aware IPv6 over IPv4 Tunnels

You can forward IPv6 packets on an untrusted IPv4 infrastructure by creating Virtual Routing and Forwarding (VRF)-aware IPv6 tunnels in it. These tunnels can have endpoints in a VRF table or in a global routing table. The tunnel modes used are **tunnel mode gre ipv4** (default mode) and **tunnel mode ipv4**.

How to Configure VRF-Aware IPv6 Tunnels

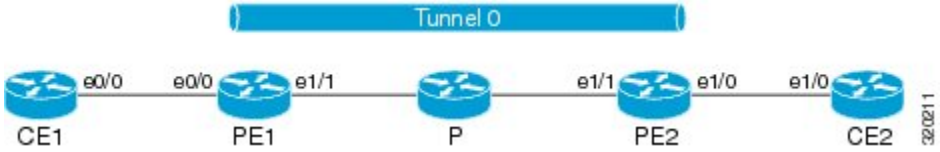
To configure a VRF-aware tunnel, you need to perform the following steps:

1. **Define the customer and transport VRF**—Define a customer VRF if the tunnel is VRF-aware. Define a transport VRF if the tunnel endpoint needs to be in a VRF. See the Defining a VRF Instance, on page 3184 section.
2. **Set up the network**—Configure relevant interfaces and configure relevant routes. Ensure that a valid route exists between the PE devices and the PE device and the customer network.
3. **Configure the tunnel between the PE devices**—See theConfiguring a VRF-Aware Tunnel, on page 3181 section.
   a. **Configure the tunnel address**
   b. **Configure the tunnel source**—This is an interface on the PE device.
   c. **Configure the tunnel destination**—This is tunnel source of the other PE device. For proper configuration of the tunnel, ensure that the tunnel destination is reachable from the PE device with a ping command (A valid route must exist to the tunnel destination).
   d. **Configure the tunnel mode**
4. **Configure customer edge network**. See theConfiguring Customer Edge Networks for Tunneling , on page 3185 section.
5. **Configure static routes using the tunnel**—Configure routes on the PE devices to remote CE networks using the configured tunnel.

Configuring a VRF-Aware Tunnel

This task configures a tunnel between PE1 and PE2, as shown in the image below. The configuration task need to be repeated on both PE devices, PE1 and PE2.

Figure 218: Configuring a VRF-Aware Tunnel



SUMMARY STEPS

1. **interface** *type number*
2. **vrf forwarding** *transport-vrf-name*
3. • **ip address** *ip-address mask* or
   • **ipv6 address** *ipv6-address/prefix-length*
4. **exit**
5. Configure static routes between provider edge devices.
6. **interface tunnel** *number*
7. **vrf forwarding** *customer-vrf-name*
8. • **ip address** *ip-address mask* or
   • **ipv6 address** *ipv6-address/prefix-length*
9. **tunnel source** *interface-type interface-number*
10. **tunnel destination** [*ip-address* | *ipv6-address*]
11. **tunnel vrf** *transport-vrf-name*
12. **tunnel mode** {**aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ipv6 | ipip [decapsulate-any] | ipsec ipv4 | iptalk | ipv6 | ipsec ipv6 | mpls | nos | rbscp**}
13. **exit**
14. • **ip route** [**vrf** *vrf-name*] *prefix mask interface-type interface-number* [*next-hop-ip-address*] or
    • **ipv6 route** [**vrf** *vrf-name*] *destination-ipv6-prefix interface-type interface-number* [*next-hop-ipv6-address*]
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Device(config)# interface ethernet 1/1`	Configures the interface used as a tunnel source.
Step 2	Required: **vrf forwarding** *transport-vrf-name* **Example:** `Device(config-if)# vrf forwarding red`	(Optional) Associates the transport VRF with the tunnel. **Note** This step is not required if the tunnel endpoints are in the global routing table.

Step 3	• **ip address** *ip-address mask* or
	• **ipv6 address** *ipv6-address/prefix-length*
	Example:
	`Device(config-if)# ip address 10.22.22.22 255.255.255.255`
	or
	`Device(config-if)# ipv6 address 2001:DB8:3::1/64`
		Sets an IP address for the tunnel source interface.
		• The address configured in this step for PE1 is used as the tunnel endpoint or tunnel destination while configuring the tunnel on PE2 and vice versa.
		• This address may be in the global routing table or in the VRF.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 5	Configure static routes between provider edge devices.	Provider edge devices are reachable with a **ping** or **ping vrf** command.
Step 6	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 0`	Configures the tunnel interface and enters interface configuration mode. The same tunnel needs to be configured on PE2.
Step 7	**vrf forwarding** *customer-vrf-name* **Example:** `Device(config-if)# vrf forwarding green`	(Optional) Associates the customer VRF instance with the tunnel. • Packets exiting the tunnel are forwarded to this VRF (inner IP packet routing). **Note** This step is required only for VRF-aware tunnels.

Step 8	• **ip address** *ip-address mask* or
	• **ipv6 address** *ipv6-address/prefix-length*
	Example:
	`Device(config-if)# ip address 10.4.1.1 255.255.255.0`
	or
	`Device(config-if)# ipv6 address 2001:DB8:3::1/64`
		Configures an IPv4 or IPv6 address for the tunnel.
		• This address is used as the next-hop address while configuring static routes. Ensure that PE1 and PE2 have addresses within the same network.

Step 9	**tunnel source** *interface-type interface-number* **Example:** `Device(config-if)# tunnel source ethernet 1/1`	Sets the source address for a tunnel interface.
Step 10	**tunnel destination** [*ip-address* | *ipv6-address*] **Example:** `Device(config-if)# tunnel destination 10.44.44.44`	(Optional) Specifies the destination for a tunnel interface. • The tunnel source address of device PE2 is used as the tunnel destination address of PE1 and vice versa. • If an IPv6 infrastructure exists between the two PE devices, use an IPv6 address. If an IPv4 infrastructure exists between the two PE devices, use an IPv4 address (IPv6 over IPv4 tunnel).
Step 11	**tunnel vrf** *transport-vrf-name* **Example:** `Device(config-if)# tunnel vrf red`	(Optional) Associates the transport VRF with the tunnel. • This VRF is the same as the VRF associated with the physical interface over which the tunnel sends packets (outer IP packet routing).

	Command or Action	Purpose
		Note This step is not required if the tunnel endpoints are in the global routing table.
Step 12	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip [decapsulate-any]** | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**} **Example:** `Device(config-if)# tunnel mode ipv6`	(Optional) Sets the encapsulation mode for the tunnel interface. **Note** This step is not required if the tunnel mode is GRE IPv4 as this is the default mode.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

Step 14	• **ip route** [**vrf** *vrf-name*] *prefix mask interface-type interface-number* [*next-hop-ip-address*] or
	• **ipv6 route** [**vrf** *vrf-name*] *destination-ipv6-prefix interface-type interface-number* [*next-hop-ipv6-address*]
	Example:
	`Device(config)# ip route 10.44.44.0 255.255.255.0 10.22.22.23`
	`Device(config)# ip route vrf red 10.44.44.0 255.255.255.0 10.22.22.23`
	or
	`Device(config)# ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:1::2`
	`Device(config)# ipv6 route vrf green 2001:DB8:2:2::/64 2001:DB8:2:1::2`
		Establishes static routes to remote customer networks by using the configured tunnel.
		• Use the tunnel address as the next hop.
		• For PE1, configure a static route to network PE2-CE2. For PE2, configure a static route to network PE1-CE1.

Step 15	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

What to do next

Verify the IPv6 Tunnels. See Verifying VRF-Aware Tunnels , on page 3186

Defining a VRF Instance

Perform this task to make a device Virtual Routing and Forwarding (VRF)-aware and to configure VRF-aware tunnels.

SUMMARY STEPS

1. **vrf definition** *vrf-name*
2. **rd** *route-distinguisher*
3. **route-target export** *route-target-ext-community*

4. **route-target import** *route-target-ext-community*
5. **address-family {ipv4 | ipv6}**
6. **exit-address-family**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition green`	Enters IP VRF configuration mode for defining a VRF routing table instance.
Step 2	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:1`	Specifies a route distinguisher (RD) for a VRF instance.
Step 3	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target export 1:1`	Exports routing information to the target VPN extended community.
Step 4	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf)# route-target import 1:1`	Imports routing information to the target VPN extended community.
Step 5	**address-family {ipv4 | ipv6}** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address-family configuration mode to configure a routing session using standard IPv4 or IPv6 address prefixes.
Step 6	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address-family configuration mode and enters IP VRF configuration mode.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits IP VRF configuration mode and enters global configuration mode.

Configuring Customer Edge Networks for Tunneling

Perform this task to configure a customer edge (CE) network. In this configuration, the CE network is a network with CE devices connected to a provider edge (PE) device. PE1 and CE1 are connected and PE2 and CE2 are connected. Addresses must be configured accordingly.

Before you begin

To define a customer VRF, see the Defining a VRF Instance, on page 3184 section.

SUMMARY STEPS

1. **interface** *type number*
2. **vrf forwarding** *customer-vrf-name*
3. • **ip address** *ip-address mask* or
   • **ipv6 address** *ipv6-address/prefix-length*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Device(config)# interface Ethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 2	**vrf forwarding** *customer-vrf-name* **Example:** `Device(config-if)# vrf forwarding green`	(Optional) Associates a VRF instance or a virtual network with the tunnel. **Note** This step is required only if the interface needs to be associated with a VRF.

Step 3	• **ip address** *ip-address mask* or
	• **ipv6 address** *ipv6-address/prefix-length*
	Example:
	`Device(config-if)# ip address 10.22.22.22 255.255.255.0`
	or
	`Device(config-if)# ipv6 address 2001:DB8:1::1/64`
		Configures an address for the interface.
		• Ensure that CE devices connected to the PE device are on the same network.

Step 4	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

Verifying VRF-Aware Tunnels

Use the following commands to verify Virtual Routing and Forwarding (VRF)-aware tunnels:

SUMMARY STEPS

1. **show tunnel interface**
2. **show ip route** *ip-address*
3. **show ip route vrf** *vrf-name ip-address*
4. **ping ipv6** *ipv6-address* **source** *ipv6-address*
5. **ping vrf** *vrf-name ipv6-address* **source** *ipv6-address*
6. **debug ipv6 icmp**

DETAILED STEPS

Step 1 **show tunnel interface**

This command displays detailed information about all tunnel interfaces.

Example:

The following is sample output from a provider edge (PE) device with Generic Routing Encapsulation (GRE) tunnel mode:

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source Loopback2
   IP transport: output interface Ethernet1/0 next hop 10.0.0.2,
   Tunnel header destination 10.44.44.44
   Application ID 1: unspecified
   Linestate - current up, cached up
   Internal linestate - current up, evaluated up
```

Example:

The following is sample output from a PE device with IPv6/IP tunnel mode:

```
Device# show tunnel interface

Tunnel0
   Mode:IPv6/IP, Destination 44.44.44.44, Source Loopback2
   IP transport: output interface Ethernet1/0 next hop 2.0.0.2,
   Tunnel header destination 44.44.44.44
   Application ID 1: unspecified
   Linestate - current up, cached up
   Internal linestate - current up, evaluated up
```

The output is displayed and the tunnel mode is observed.

Step 2 **show ip route** *ip-address*

This command displays detailed routing information to a tunnel destination address.

Example:

The following is sample output from a PE device with the tunnel endpoint in the global routing table:

```
Device# show ip route 10.44.44.44

  Routing entry for 10.44.44.44/32
  Known via "ospf 1", distance 110, metric 21, type intra area
  Last update from 10.0.0.2 on Ethernet1/0, 01:10:25 ago
  Routing Descriptor Blocks:
  * 10.0.0.2, from 10.44.44.44, 01:10:25 ago, via Ethernet1/0
      Route metric is 21, traffic share count is 1
```

The following is sample output from a PE device having tunnel endpoints in the VRF table:

```
Device# show ip route 10.44.44.44

% Network not in table
```

The output is displayed and you can observe if the tunnel destination is in the global routing table or not.

Step 3 **show ip route vrf** *vrf-name ip-address*

This command displays detailed routing information to a destination IP address.

Example:

The following is sample output from PE1:

```
Device# show ip route vrf green 10.4.4.4

Routing entry for 10.4.4.4/32
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.0.0.2, via Ethernet1/0
      Route metric is 0, traffic share count is 1
```

The tunnel destination address 10.4.4.4 is not in the global routing table.

Step 4 **ping ipv6** *ipv6-address* **source** *ipv6-address*

This command displays the status of the connectivity between two devices.

Example:

The following is sample output from a customer edge (CE) device CE1 with a **ping** command issued to CE2:

```
Device# ping ipv6 2001:DB8:2::1 source 2001:DB8:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/2/4 ms
```

Step 5 **ping vrf** *vrf-name ipv6-address* **source** *ipv6-address*

The VRF-ping tests the VPN connection.

Example:

The following is sample output from CE1 with a **ping vrf** command issued to CE2:

```
Device# ping vrf green ipv6 2001:DB8:2::1 source 2001:DB8:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:1::2%green
!!!!!
```

If the displayed output indicates success, the VPN is configured correctly.

Step 6 **debug ipv6 icmp**

This command displays debugging messages for IPv6 Internet Control Message Protocol (ICMP) transactions.

Example:

The following is sample output:

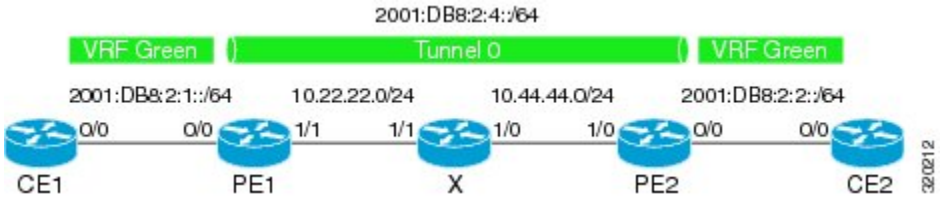
```
Device# debug ipv6 icmp

  ICMP Packet debugging is on

*Apr  6 14:08:10.743: ICMPv6: Received echo request, Src=2001:DB8:1::2, Dst=2001:DB8:2::1
```

```
*Apr  6 14:08:10.743: ICMPv6: Sent echo reply, Src=2001:DB8:2::1, Dst=2001:DB8:1::2
...
```

If the displayed output indicates success, the VPN is configured correctly.

Configuration Examples for VRF-Aware Tunnels

Example: Configuring a VRF-Aware Tunnel (Tunnel Endpoint in Global Routing Table)



Example: Configuring CE1

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:2::/64 2001:DB8:2:1::2
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:1::2
!
```

Example: Configuring PE1

```
ipv6 unicast-routing
ipv6 cef
!
```

```
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Tunnel0
 no ip address
 vrf forwarding green
 ipv6 address 2001:DB8:2:4::1/64
 tunnel source 10.22.22.22
 tunnel destination 10.44.44.44
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.22 255.255.255.0
 no shutdown
 exit
!
ip route 10.44.44.0 255.255.255.0 10.22.22.23
ipv6 route vrf green 2001:DB8:2:2::/64 Tunnel0 2001:DB8:2:4::2
```

Example: Configuring PE2

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Tunnel0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 tunnel source 10.44.44.44
 tunnel destination 10.22.22.22
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
```

```
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.44 255.255.255.0
 no shutdown
 exit
!
ip route 10.22.22.0 255.255.255.0 10.44.44.43
!
ipv6 route vrf green 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:4::1
!
```

Example: Configuring CE2

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:1::/64 2001:DB8:2:2::1
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:2::1
!
```

Example: Configuring Device X

```
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.43 255.255.255.0
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.23 255.255.255.0
 no shutdown
 exit
!
```

Example: Verifying the Tunnel Configuration

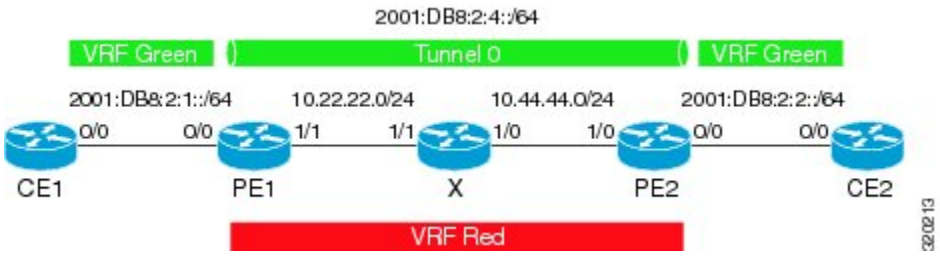
From CE1

```
Device# ping vrf green ipv6 2001:db8:2:2::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/6 ms

Device# ping vrf green ipv6 2001:db8:2:2::2 source 2001:db8:2:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:2:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

From PE1

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source 10.22.22.22
   IP transport: output interface Ethernet1/1 next hop 10.22.22.23
   Application ID 1: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up
   Tunnel Source Flags: Local
   Transport IPv4 Header DF bit cleared
   OCE: IP tunnel decap
   Provider: interface Tu0, prot 47
     Performs protocol check [47]
     Protocol Handler: GRE: opt 0x0
       ptype: ipv4 [ipv4 dispatcher: punt]
       ptype: ipv6 [ipv6 dispatcher: from if Tu0]
       ptype: mpls [mpls dispatcher: drop]
       ptype: otv [mpls dispatcher: drop]
       ptype: generic [mpls dispatcher: drop]
There are 0 tunnels running over the EON IP protocol
There are 0 tunnels running over the IPinIP protocol
There are 0 tunnels running over the NOSIP protocol
There are 0 tunnels running over the IPv6inIP protocol
There are 0 tunnels running over the RBSCP/IP protocol


Device# show ip route 10.44.44.44

Routing entry for 10.44.44.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.22.22.23
      Route metric is 0, traffic share count is 1

Device# debug ipv6 icmp

  ICMP Packet debugging is on
*Jan  1 10:57:37.882: ICMPv6: Sent R-Advert, Src=FE80::A8BB:CCFF:FE00:5200, Dst=FF02::1
*Jan  1 11:00:18.634: ICMPv6: Received R-Advert, Src=FE80::A8BB:CCFF:FE00:5200,Dst=FF02::1
```

Example: Configuring a VRF-Aware Tunnel (Tunnel Endpoint in VRF)



Example: Configuring CE1

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:2::/64 2001:DB8:2:1::2
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:1::2
!
```

Example: Configuring PE1

```
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
vrf definition red
rd 2:2
route-target export 2:2
```

```
route-target import 2:2
address-family ipv4
exit-address-family
exit
!
interface Tunnel0
 no ip address
 vrf forwarding green
 ipv6 address 2001:DB8:2:4::1/64
 tunnel source 10.22.22.22
 tunnel destination 10.44.44.44
 tunnel vrf red
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
interface Ethernet1/1
 vrf forwarding red
 no ip address
 ip address 10.22.22.22 255.255.255.0
 no shutdown
 exit
!
ip route vrf red 10.44.44.0 255.255.255.0 10.22.22.23
ipv6 route vrf green 2001:DB8:2:2::/64 Tunnel0 2001:DB8:2:4::2
```

Example: Configuring PE2

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
vrf definition red
rd 2:2
route-target export 2:2
route-target import 2:2
address-family ipv4
exit-address-family
exit
!
interface Tunnel0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 tunnel source 10.44.44.44
 tunnel destination 10.22.22.22
 tunnel vrf red
```

```
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 vrf forwarding red
 no ip address
 ip address 10.44.44.44 255.255.255.0
 no shutdown
 exit
!
ip route vrf red 10.22.22.0 255.255.255.0 10.44.44.43
!
ipv6 route vrf green 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:4::1
!
```

Example: Configuring CE2

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:1::/64 2001:DB8:2:2::1
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:2::1
!
```

Example: Configuring Device X

```
!
interface Ethernet1/0
 vrf forwarding red
 no ip address
 ip address 10.44.44.43 255.255.255.0
 no shutdown
 exit
!
```

```
interface Ethernet1/1
 vrf forwarding red
 no ip address
 ip address 10.22.22.23 255.255.255.0
 no shutdown
 exit
!
```

Example: Verifying the Tunnel Configuration

From CE1

```
Device# ping vrf green ipv6 2001:db8:2:2::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/6 ms

Device# ping vrf green ipv6 2001:db8:2:2::2 source 2001:db8:2:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:2:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

From PE1

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source 10.22.22.22
   IP transport: output interface Ethernet1/1 next hop 10.22.22.23
   Application ID 1: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up
   Tunnel Source Flags: Local
   Transport IPv4 Header DF bit cleared
   OCE: IP tunnel decap
   Provider: interface Tu0, prot 47
     Performs protocol check [47]
     Protocol Handler: GRE: opt 0x0
       ptype: ipv4 [ipv4 dispatcher: punt]
       ptype: ipv6 [ipv6 dispatcher: from if Tu0]
       ptype: mpls [mpls dispatcher: drop]
       ptype: otv [mpls dispatcher: drop]
       ptype: generic [mpls dispatcher: drop]
There are 0 tunnels running over the EON IP protocol
There are 0 tunnels running over the IPinIP protocol
There are 0 tunnels running over the NOSIP protocol
There are 0 tunnels running over the IPv6inIP protocol
There are 0 tunnels running over the RBSCP/IP protocol

Device# show ip route 10.44.44.44

% Network not in table

Device# show ip route vrf red 10.44.44.44
```

```
Routing Table: red
Routing entry for 10.44.44.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.22.22.23
      Route metric is 0, traffic share count is 1

Device# debug ipv6 icmp

  ICMP Packet debugging is on
*Jan  1 10:57:37.882: ICMPv6: Sent R-Advert, Src=FE80::A8BB:CCFF:FE00:5200, Dst=FF02::1
*Jan  1 11:00:18.634: ICMPv6: Received R-Advert, Src=FE80::A8BB:CCFF:FE00:5200,Dst=FF02::1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*Cisco IOS IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Standards and RFCs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRF-Aware Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 325: Feature Information for VRF-Aware Tunnels

Feature Name	Releases	Feature Information
VRF-Aware Tunnels	Cisco IOS XE Release 3.8S	Virtual Routing and Forwarding (VRF)-aware tunnels are used to connect customer networks separated by untrusted core networks or core networks with different infrastructures (IPv4 or IPv6). The following command was modified to support IPv6 transport: **tunnel vrf**.

CHAPTER 268

Ethernet over GRE Tunnels

The Ethernet over GRE Tunnels feature allows customers to leverage existing low–end residential gateways to provide mobility services to mobile nodes using Proxy Mobile IPv6 (PMIPv6), General Packet Radio Service (GPRS) Tunneling Protocol (GTP), and Intelligent Service Gateway (ISG).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Ethernet over GRE Tunnels

- Mobile nodes can have only IPv4 addresses
- IPv6 mobile clients are not supported
- If the VLAN priority tag inside the EoGRE packet is set to a nonzero value, ISG or iWAG ignores the packet

Information About Ethernet over GRE Tunnels

The Ethernet over GRE tunnels feature allows customers to leverage existing low-end residential gateways to provide mobility services to mobile nodes.

As service provider Wi-Fi space gains popularity, Cisco customers need to provide access to the Internet and mobile services using public hotspots. A high–end RG can provide these mobility services using Proxy Mobile IPv6 (PMIPv6), Intelligent Service Gateway (ISG) or General Packet Radio Service (GPRS) Tunneling Protocol (GTP).

Low-end RGs or customer premises equipment (CPE) can be used to forward traffic from Mobile nodes to high-end devices. These RGs or CPE can be configured in bridged mode, and Ethernet over Generic Routing Encapsulation (GRE) tunnels can be used to forward Ethernet traffic to the aggregation device.

Mobile nodes access the Internet over Wi-Fi access points (APs). The APs are either autonomous or connected to a wireless LAN controller (WLC). These APs and WLCs are generically referred to as RGs or CPEs. The CPEs are located at individual or community residences and may be connected to the service–provider network through a connection mechanism like an asymmetric DSL (ADSL) modem or a cable modem. The connection mechanism is transparent to the aggregation device.

These CPEs are provided, provisioned, and managed by the service provider as a part of the broadband access service. Generally, there is extra bandwidth on the Wi-FI AP as well as the back-end pipe to the service provider, which can be used to provide mobile–Internet services to roaming customers in the vicinity.

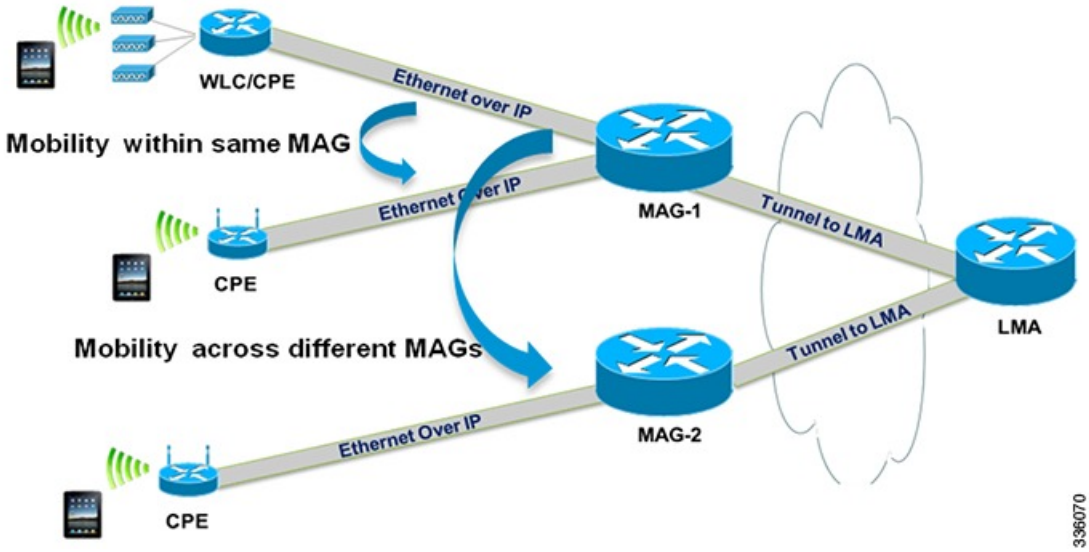
Mobility Services Using PMIPv6

You can use PMIPv6 to provide mobility services to mobile devices, but you would require high-end RGs with Mobile Access Gateways (MAG) functionality.

RGs or CPEs can also be used to forward traffic from Mobile nodes to MAG-enabled aggregation devices using Ethernet over GRE tunnels.

The aggregation device can create IP sessions and allocate IP addresses (locally or in proxy mode) in a manner similar to regular IP sessions on physical Ethernet interfaces.

Figure 219: Mobility Services Using PMIPv6



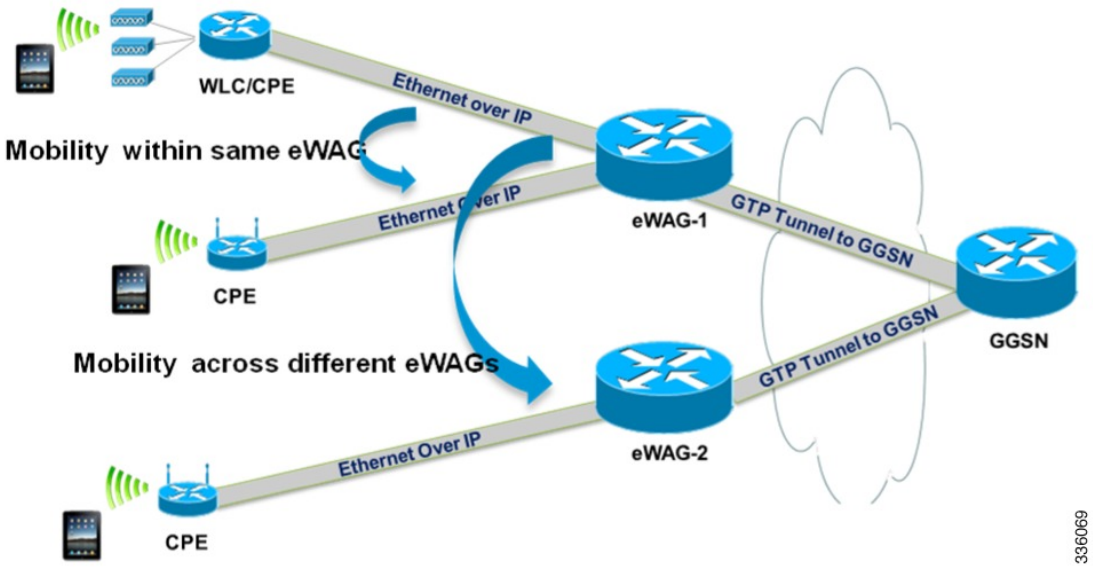
In the deployment scenario given in the above figure, MAG-1 and MAG-2 are configured to handle tunneled Ethernet traffic from access side and also have regular IP tunnels to one or more local mobility anchor (LMA).

Mobility Services Using GTP

You can use GTP to provide mobility services to mobile devices, but you would require high-end RGs with Enhanced Wireless Access Gateway functionality.

RGs or CPEs can also be used to forward traffic from Mobile nodes to Enhanced Wireless Access Gateway devices using Ethernet over GRE tunnels.

Figure 220: Mobility Services Using GTP



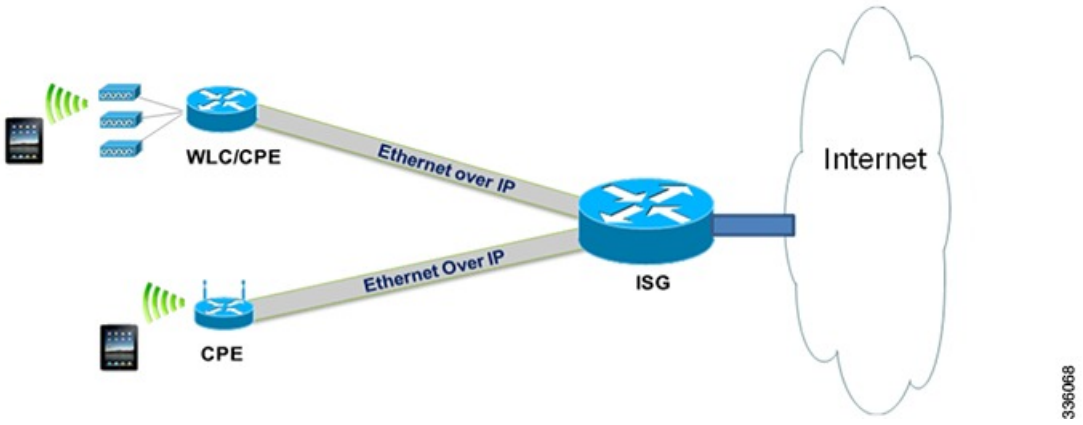
In the deployment scenario given in the above figure, eWAG-1 and eWAG-2 are configured to handle tunneled Ethernet traffic from access side and also have one or more GTP tunnels to one or more gateway Cisco General packet radio service (GPRS) support node (GGSN) devices.

Mobility Services Using ISG

You can use ISG to provide simple IP services to mobile devices but you would require a high-end RGs with ISG functionality.

RGs or CPEs can also be used to forward traffic from Mobile nodes to ISG devices using Ethernet over GRE tunnels as shown in the figure below.

Figure 221: Mobility Services Using ISG



Ethernet over GRE Tunnels Supported Functionality

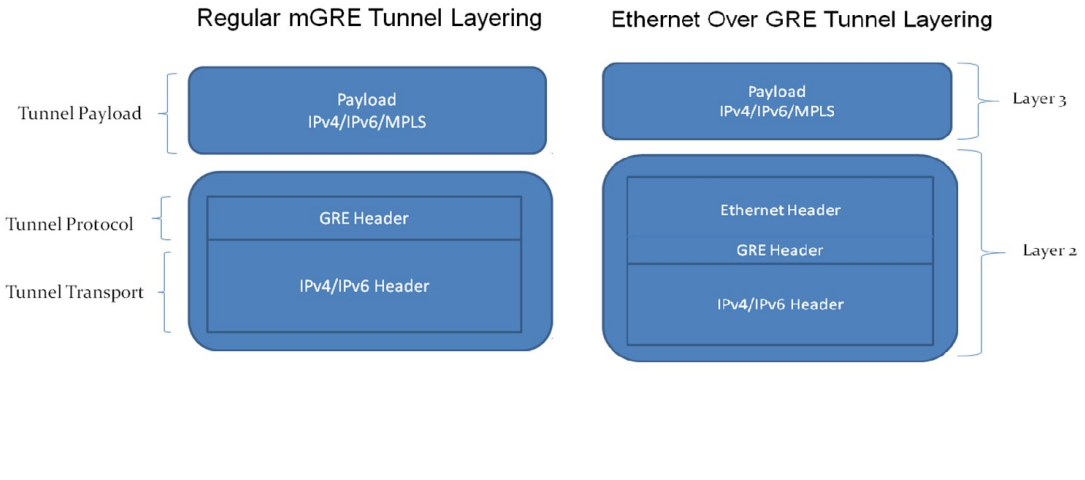
The Ethernet over GRE tunnels feature supports the following functionality:

- Mobility services can be provided to the mobile nodes using existing low-end residential gateways (RGs) using Ethernet over generic routing and encapsulation (GRE) tunnels. Intelligent Service Gateway (ISG), Proxy Mobile IPv6 (PMIPv6), and GPRS Tunneling Protocol (GTP) can be used to provide the mobility services.
- Ethernet frames can be transported over IPv6 and IPv4 infrastructures. Customer premises Equipment (CPE) is pre–configured with a point-to-point Generic Routing Encapsulation (GRE) IPv4 or IPv6 tunnel. The tunnel destination is a well-known IPv4 or IPv6 address of an aggregation device.
- Tunnels can be configured to be part of a single VLAN—The CPE may insert a VLAN tag in the Ethernet frame. Only a single VLAN tag is supported.
- Tunnels can be configured with a statically configured, symmetric GRE key. You can use the **tunnel key** command to configure this key.
- Sessions can be created with DHCP for IPv4 (DHCPv4), unclassified MAC, and Address Resolution Protocol (ARP) Detecting Network Attachments for IPv4 (DNAv4).

Tunnel Encapsulation in Ethernet over GRE tunnels

Tunnel encapsulation in Ethernet over GRE tunnels is similar to tunnel encapsulation in multipoint Generic Routing Encapsulation (mGRE) tunnels, given in the below figure.

Figure 222: Comparison of Ethernet over GRE tunnels and mGRE tunnels



The mGRE tunnel is a nonbroadcast multiAccess (NBMA) interface that can handle multiple tunnel endpoints. The mGRE tunnel can forward payloads like IPv4, IPv6, and Multiprotocol Label Switching (MPLS) in GRE–encapsulated IPv4/IPv6 transport frames from different endpoints, which can then be sent to specific endpoints. While transmitting, the mGRE tunnel interface encapsulates the payload with GRE and transports IPv4/IPv6 headers. On the receiving end, the mGRE tunnel interface strips the GRE and transport header and forwards the payload.

In Ethernet over GRE tunnels, the Ethernet header is included in the tunnel encapsulation along with GRE and transport header.

The tunnel modes used for Ethernet over GRE IPv4 transport can be set using the **tunnel mode ethernet gre ipv4** command.

Similarly, the tunnel modes used for Ethernet over GRE IPv6 transport can be set using the **tunnel mode ethernet gre ipv6** command.

You can see the source of the tunnel by using the **show tunnel source tracking** command.

Although the Ethernet over GRE tunnel simulates regular Ethernet behavior for all practical purposes, the interface is an NBMA interface at the data-link layer. As there may be many mobile nodes and CPE connected to the Ethernet over GRE tunnel, broadcasting a packet is not supported. Even if an aggregation device like the Mobile Access Gateway (MAG) needs to use a broadcast MAC address in the downstream packet frame, the message is unicast to only the respective CPE. Similarly, multicast messages are also sent as unicast messages to the mobile nodes.

Virtual MAC Address

An Ethernet over GRE tunnel is configured with a virtual MAC address. When a packet enters the tunnel, the tunnel accepts the packet only if the destination MAC address of the packet matches the virtual MAC address of the tunnel or the broadcast MAC address. Otherwise, the packet is dropped.

Note If the tunnel interface is configured to handle multicast traffic for specific multicast groups, the corresponding MAC addresses are also accepted by the tunnel.

If PMIPv6 or GTP is enabled on the tunnel, the protocols provide a virtual MAC address that is used as the source MAC address of packets exiting the tunnel. If PMIPv6 or GTP is not enabled, the virtual MAC address of the tunnel interface is used as the source MAC address of the exiting packets.

Virtual MAC addresses are associated with the tunnel using the **mac-address** command. You can use the **show tunnel mac-table** command to see MAC table entries. You can use the **test tunnel mac-address** command to test the addition of MAC addresses to the MAC table of a tunnel interface.

VLAN on the Tunnel Interface

Mobile nodes connect to the wireless access points (APs). These APs have Service Set Identifiers (SSIDs) provided by the service provider. The SSID of a CPE is the VLAN identifier. The CPE can be configured to insert VLAN tags in Ethernet frames received from the mobile nodes before forwarding them on the GRE tunnel. Similarly, for downstream traffic, the GRE tunnel can be configured to insert a VLAN tag in all Ethernet frames sent to the MN.

A tunnel interface supports only one VLAN tag.

You can associate a VLAN with an Ethernet over GRE tunnel by using the **tunnel vlan** command.

How to Configure an Ethernet over GRE tunnel

Configuring an Ethernet over GRE Tunnel

SUMMARY STEPS

1. **interface tunnel** *tunnel-number*

2. **mac-address** *mac-address*
3. Do one of the following:
   - **ip address dhcp**
   - **ip address** *ip-address mask*
4. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
5. **tunnel mode ethernet gre** {**ipv4 | ipv6**}
6. **tunnel key** *key*
7. **tunnel vlan** *vlan-id*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 2	**mac-address** *mac-address* **Example:** `Device(config-if)# mac-address 0000.0000.0001`	(Optional) Specifies a MAC address for the tunnel.

Step 3	Do one of the following:
	• **ip address dhcp**
	• **ip address** *ip-address mask*
	Example:
	`Device(config-if)# ip address 192.168.4.3 255.255.255.0`
	Example:
	`Device(config-if)# ip address dhcp`
		• Specifies that the IP address of the mobile node is allocated by DHCP when it connects to the network.
		• Specifies the IPv4 address of the mobile node.

Step 4	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Device(config-if)# tunnel source loopback 2`	Sets the source address of a tunnel interface.
Step 5	**tunnel mode ethernet gre** {**ipv4 | ipv6**} **Example:** `Device(config-if)# tunnel mode ethernet gre ipv4`	Sets the encapsulation mode of the tunnel to Ethernet over GRE IPv4 or GRE IPv6.
Step 6	**tunnel key** *key* **Example:** `Device(config-if)# tunnel key 1`	Enables an key identifier for the tunnel interface.

	Command or Action	Purpose
Step 7	**tunnel vlan** *vlan-id* **Example:** `Device(config-if)# tunnel vlan 1`	Associates a VLAN identifier with the Ethernet over GRE tunnel.
Step 8	**end** **Example:** `end`	Exits to privileged EXEC mode.

What to do next

Verify the tunnel.

Verifying Ethernet Over GRE Tunnel

Before you begin

Configure the Ethernet over GRE tunnel.

SUMMARY STEPS

1. **show interface tunnel**
2. **show tunnel mac-table**
3. **show tunnel endpoints**

DETAILED STEPS

Step 1 **show interface tunnel**

This command displays information about the tunnel.

Example:

```
Device# show interface tunnel 1

Tunnel1 is up, line protocol is up
Hardware is Tunnel
Internet address is 11.1.1.1/24
MTU 17846 bytes, BW 100 Kbit/sec, DLY 50000 usec, reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.0.1
Tunnel MAC address 0000.5e00.5213
Tunnel Vlan-id 1
Tunnel protocol/transport Ethernet-GRE/IP Key 0x1, sequencing disabled Checksumming of packets disabled
Tunnel TTL 255
Tunnel transport MTU 1454 bytes
Tunnel transmit bandwidth 8000 (kbps) Tunnel receive bandwidth 8000 (kbps)
Last input 00:48:08, output never, output hang never
Last clearing of "show interface" counters 00:48:26
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 107
Queueing strategy: fifo
Output queue: 0/0 (size/max)
```

```
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
1867 packets input, 161070 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
43 packets output, 4386 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out ind-uut#
--- 22:03:51 ---
44: 2013-01-30T22:03:51: %SCRIPT-6-INFO: {_haExecCmd: Executing cmd exec with ind-uut-a}


Device# show interface tunnel 2

Tunnel2 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.1.1.1/24
MTU 1434 bytes, BW 100 Kbit/sec, DLY 50000 usec, reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10::1
Tunnel MAC address 0000.5e00.5213
Tunnel Vlan-id 2
Tunnel protocol/transport Ethernet-GRE/IPv6
Key 0x2, sequencing disabled
Checksumming of packets disabled
Tunnel TTL 255
Path MTU Discovery, ager 10 mins, min MTU 1280
Tunnel transport MTU 1434 bytes
Tunnel transmit bandwidth 8000 (kbps) Tunnel receive bandwidth 8000 (kbps)
Last input never, output never, output hang never
Last clearing of "show interface" counters 00:48:28
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 106
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
```

Step 2 show tunnel mac-table

This command displays MAC table entries associated with a tunnel.

Example:

```
Device# show tunnel mac-table tunnel0

CPE IP 1.1.1.1 Refcount 2 Base 0x2A98DD0000
    mac-address 0122.0111.0111 vlan 1
    mac-address 0011.1111.0001 vlan 2
CPE IP 3.3.3.3 Refcount 2 Base 0x12345678
   mac-address 1234.5678.9011 vlan 1
```

Step 3 show tunnel endpoints

This command displays tunnel endpoints and verifies if the tunnel has been created correctly.

Example:

```
Device# show tunnel endpoints

Tunnel0 running in Ethernet-GRE/IP mode

Endpoint transport 10.1.1.1 Refcount 3 Base 0x2A98DD03C0 Create Time 3d02h
   overlay 10.1.1.1 Refcount 2 Parent 0x2A98DD03C0 Create Time 3d02h
 Endpoint transport 3.3.3.3 Refcount 3 Base 0x2A98DD0300 Create Time 3d02h
   overlay 10.1.1.3 Refcount 2 Parent 0x2A98DD0300 Create Time 3d02h
```

Configuration Examples for Ethernet over GRE Tunnels

Example: Configuring Ethernet over GRE Tunnels

Configuring Ethernet over GRE tunnels on the Mobile Node

```
! Configure the topology
mobile-node1(config-if)# interface GigabitEthernet0/1
mobile-node1(config-if)# ip address 10.21.1.1 255.255.255.0
mobile-node1(config-if)# no shut
mobile-node1(config-if)# exit
mobile-node1(config)# ip route 10.0.0.1 255.255.255.255 10.21.1.2

! Configuring the interface used as the source of the tunnel
mobile-node1(config)# interface Loopback0
mobile-node1(config-if)# ip address 10.40.0.1 255.255.255.0
mobile-node1(config-if)# ipv6 address 2001:db8:2:40::1/64
mobile-node1(config-if)# no shutdown

! Configuring the Ethernet over GRE IPv4 Tunnel
mobile-node1(config-if)# interface Tunnel1
mobile-node1(config-if)# mac-address 0000.0000.0001
mobile-node1(config-if)# ip dhcp client client-id ascii MN1@cisco.com
mobile-node1(config-if)# ip address dhcp
mobile-node1(config-if)# no ip redirects
mobile-node1(config-if)# no ip route-cache
mobile-node1(config-if)# tunnel source Loopback0
mobile-node1(config-if)# tunnel mode ethernet gre ipv4
mobile-node1(config-if)# tunnel key 1
mobile-node1(config-if)# tunnel vlan 1
mobile-node1(config-if)# no shutdown
```

Configuring Ethernet over GRE tunnel on the MAG

```
! Configure the topology
MAG(config)# interface FastEthernet1/1/5
MAG(config-if)# ip address 10.21.1.2 255.255.255.0
MAG(config-if)# ipv6 address 2001:db8:2:21::2/64
```

```
MAG(config-if)# no shut
MAG(config)# ip route 10.40.0.1 255.255.255.255 10.21.1.1

! Configure the interface used as source of the tunnel
MAG(config-if)# interface Loopback0
MAG(config-if)# ip address 10.0.0.1 255.255.255.0
MAG(config-if)# no shutdown

! Configuring the Ethernet over GRE IPv4 Tunnel
MAG(config)# interface Tunnel1
MAG(config-if)# ip address 10.11.1.1 255.255.255.0
MAG(config-if)# tunnel mode ethernet gre ipv4
MAG(config-if)# tunnel source 10.0.0.1

! Configuring a static GRE and VLAN ID for the tunnel
MAG(config-if)# tunnel key 1
MAG(config-if)# tunnel vlan 1

! Associating the service policy control with the tunnel
MAG(config-if)# service-policy type control DHCP1

! Enable ISG on the tunnel
MAG(config-if)# ip subscriber l2-connected
MAG(config-subscriber)# initiator unclassified mac-address
Please unconfigure existing command before configuring.
MAG(config-subscriber)# initiator dhcp class-aware
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*Cisco IOS IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Standards and RFCs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Ethernet over GRE Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 326: Feature Information for Ethernet over GRE Tunnels

Feature Name	Releases	Feature Information
Ethernet over GRE Tunnels	Cisco IOS XE Release 3.9S	The Ethernet over GRE tunnels feature allows customers to leverage existing low–end residential gateways to provide mobility services to mobile nodes using Proxy Mobile IPv6 (PMIPv6), GPRS Tunneling Protocol (GTP) and Intelligent Service Gateway (ISG). The following command was modified to add the Ethernet over GRE tunnel mode for IPv4 and IPv6: **tunnel mode ethernet gre**. The following commands were introduced:**tunnel vlan**, **show tunnel mac-table**, **show tunnel source tracking**, **test tunnel mac-address**.

CHAPTER 269

QoS on Ethernet over GRE Tunnels

The QoS on Ethernet over GRE (EoGRE) Tunnels feature enables service providers to configure one common Quality of Service (QoS) policy for all endpoints, where an end-point can be a customer premise equipment (CPE) or a VLAN on a CPE. This feature supports high availability on a route processor.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

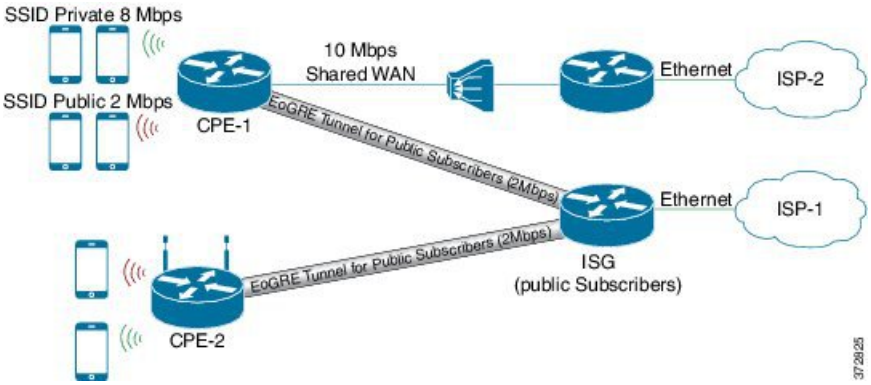
Information About QoS on Ethernet over GRE Tunnels

EoGRE Downstream QoS

The Quality of Service (QoS) on Ethernet over GRE (EoGRE) Tunnels feature enables service providers to apply a unified QoS policy on all endpoints of a tunnel. This controls the bandwidth that public subscribers can download and ensures maximum bandwidth for private customers.

In the deployment scenario given in the figure below, the total available WAN bandwidth at the customer premise equipment (CPE) is 10 Mbps, of which public users are allowed 2 Mbps and the remaining bandwidth is available for private users.

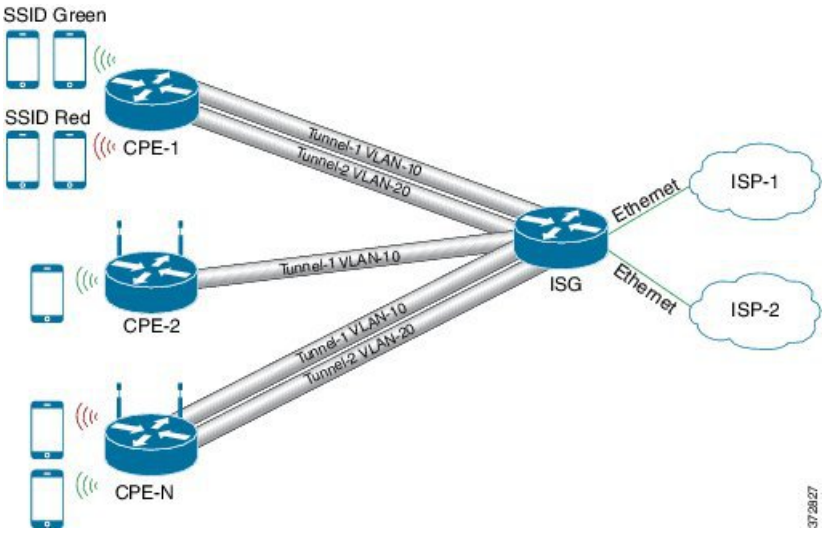
Figure 223: EoGRE Downstream QoS Use Case



Single SSID

Mobile nodes connect to wireless access points (APs). These APs have Service Set Identifiers (SSIDs) provided by the service provider. The SSID of a customer premise equipment (CPE) is the VLAN identifier. Service providers can provide more than one public SSID at a CPE. If a CPE has more than one SSID, then additional mGRE tunnels are configured with a corresponding VLAN tag. The configured multipoint generic routing encapsulation (mGRE) tunnels learn about remote subscribers and the corresponding CPEs independently. This ensures that VLANs, their subnets, default gateways, and VRFs are kept separate and independent of each other, and any QoS policy that is configured on each endpoint of these tunnels also applies to the traffic from the VLAN on the CPE.

Figure 224: Separate Tunnels for Each SSID



Multiple SSIDs

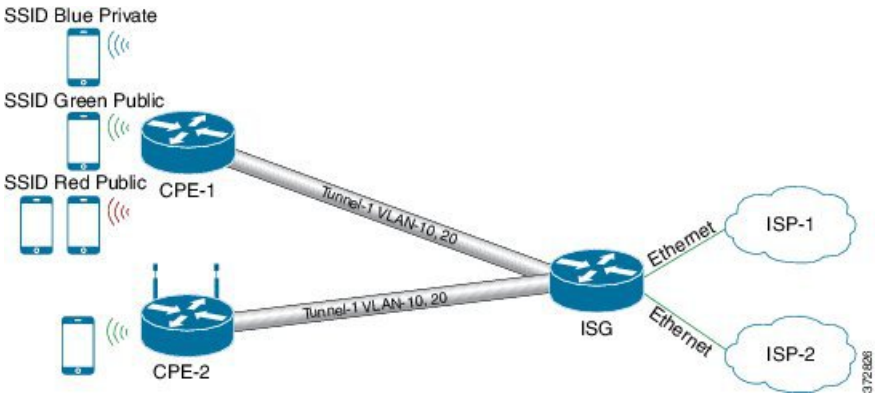
In a single tunnel for a multiple Service Set Identifiers (SSID), service providers can configure a VLAN range on the multipoint generic routing encapsulation (mGRE) tunnel. When a subscriber traffic is received, the traffic is matched according to the tunnel source and the VLAN range. The Ethernet over GRE (EoGRE)

control process also learns the MAC address of subscribers and the VLAN tag of the CPE from which the traffic is originating.

Note You cannot change a VLAN configuration if any subscriber session or MAC address is already learned in the EoGRE control process. To change the VLAN configurations, you must clear all subscriber sessions.

In the figure below, all endpoints learned on Tunnel-1 represent a CPE and a Quality of Service (QoS) policy applied on this tunnel endpoint applies to all traffic going towards the CPE irrespective of the VLAN.

Figure 225: Single Tunnel for Multiple SSIDs



How to Configure QoS on Ethernet over GRE Tunnels

Configuring Downstream QoS Policy on Ethernet over GRE Tunnels

Before you begin

Create a Quality of Service (QoS) policy map to attach to the Ethernet over GRE (EoGRE) tunnel.

Note How to create a QoS policy map is not described in the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **interface source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
5. **tunnel vlan** *vlan-id*
6. **ip address** *ip-address mask*
7. **tunnel mode ethernet gre** {**ipv4** | **ipv6**}
8. **tunnel endpoint service-policy output** *policy-map-name*

9. **ip subscriber l2-connected**
10. **initiator unclassified mac-address**
11. **initiator dhcp**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 1`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**interface source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Device(config-if)# tunnel source loopback 2`	Sets the source address of a tunnel interface.
Step 5	**tunnel vlan** *vlan-id* **Example:** `Device(config-if)# tunnel vlan 10, 20`	Associates a VLAN identifier with the Ethernet over GRE tunnel.
Step 6	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.4.3 255.255.255.0`	Specifies the IP address and mask of the mobile node.
Step 7	**tunnel mode ethernet gre** {**ipv4** | **ipv6**} **Example:** `Device(config-if)# tunnel mode ethernet gre ipv4`	Sets the encapsulation mode of the tunnel to Ethernet over GRE IPv4 or GRE IPv6.
Step 8	**tunnel endpoint service-policy output** *policy-map-name* **Example:** `Device(config-if)# tunnel endpoint service-policy output tunnel-qos-policy`	Configures the QoS policy for tunnel endpoints.
Step 9	**ip subscriber l2-connected** **Example:** `Device(config-if)# ip subscriber l2-connected`	Enters IP subscriber configuration mode.

	Command or Action	Purpose
Step 10	**initiator unclassified mac-address** **Example:** `Device(config-subscriber)# initiator unclassified mac-address`	Initiates IP sessions from unclassified MAC address.
Step 11	**initiator dhcp** **Example:** `Device(config-subscriber)# initiator dhcp`	Enables IP sessions initiated by DHCP.
Step 12	**end** **Example:** `Device(config-subscriber)# end`	Exits to global configuration mode.

Verifying QoS on Ethernet over GRE Tunnels

The **show** commands can be entered in any order.

Before you begin

Configure QoS on Ethernet over GRE (EoGRE) tunnel.

SUMMARY STEPS

1. **show interface tunnel** *tunnel-interface*
2. **show tunnel endpoints tunnel** *tunnel-interface*
3. **show tunnel mac-table tunnel** *tunnel-interface*
4. **show policy-map multipoint tunnel** *tunnel-interface*

DETAILED STEPS

Step 1 **show interface tunnel** *tunnel-interface*

This command displays information about the tunnel.

Example:

```
Device# show interface tunnel 1

Tunnel1 is up, line protocol is up
Hardware is Tunnel
Internet address is 11.1.1.1/24
MTU 17846 bytes, BW 100 Kbit/sec, DLY 50000 usec, reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.0.1
Tunnel MAC address 0000.5e00.5213
Tunnel Vlan-id 1
Tunnel protocol/transport Ethernet-GRE/IP Key 0x1, sequencing disabled Checksumming of packets disabled
Tunnel TTL 255
Tunnel transport MTU 1454 bytes
```

```
Tunnel transmit bandwidth 8000 (kbps) Tunnel receive bandwidth 8000 (kbps)
Last input 00:48:08, output never, output hang never
Last clearing of "show interface" counters 00:48:26
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 107
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
1867 packets input, 161070 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
43 packets output, 4386 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out ind-uut#
--- 22:03:51 ---
44: 2013-01-30T22:03:51: %SCRIPT-6-INFO: {_haExecCmd: Executing cmd exec with ind-uut-a}


Device# show interface tunnel 2

Tunnel2 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.1.1.1/24
MTU 1434 bytes, BW 100 Kbit/sec, DLY 50000 usec, reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10::1
Tunnel MAC address 0000.5e00.5213
Tunnel Vlan-id 2
Tunnel protocol/transport Ethernet-GRE/IPv6
Key 0x2, sequencing disabled
Checksumming of packets disabled
Tunnel TTL 255
Path MTU Discovery, ager 10 mins, min MTU 1280
Tunnel transport MTU 1434 bytes
Tunnel transmit bandwidth 8000 (kbps) Tunnel receive bandwidth 8000 (kbps)
Last input never, output never, output hang never
Last clearing of "show interface" counters 00:48:28
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 106
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
```

Step 2 **show tunnel endpoints tunnel** *tunnel-interface*

This command displays tunnel interface endpoints and verifies if the tunnel is created correctly.

Example:

```
Device# show tunnel endpoints tunnel

Tunnel0 running in Ethernet-GRE/IP mode

Endpoint transport 10.1.1.1 Refcount 3 Base 0x2A98DD03C0 Create Time 3d02h
```

```
   overlay 10.1.1.1 Refcount 2 Parent 0x2A98DD03C0 Create Time 3d02h
 Endpoint transport 3.3.3.3 Refcount 3 Base 0x2A98DD0300 Create Time 3d02h
   overlay 10.1.1.3 Refcount 2 Parent 0x2A98DD0300 Create Time 3d02h
```

Step 3 **show tunnel mac-table tunnel** *tunnel-interface*

This command displays MAC table entries that are associated with a tunnel.

Example:

```
Device# show tunnel mac-table tunnel0

overlay-address 30.0.0.21, transport-address 192.168.0.50
mac-address 0000.1200.0001, vlan 400 Mac Age 3d06h

overlay-address 60.0.0.8, transport-address 120.0.40.2
mac-address 3010.e495.b058, vlan 10 Mac Age 00:01:00
```

Step 4 **show policy-map multipoint tunnel** *tunnel-interface*

This command displays the policy-map that is associated with a tunnel.

Example:

```
Device> show policy-map multipoint tunnel 1

Interface Tunnel 1 <--> 1.1.1.1
  Service-policy output: test
    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
      police:rate 300000 bps, burst 17898 bytes
      conformed 0 packets, 0 bytes;actions:transmit
      exceeded 0 packets, 0 bytes; actions:drop
        conformed 0000 bps, exceeded 0000 bps
```

Configuration Examples for QoS on Ethernet over GRE Tunnels

Example: QoS on Ethernet over GRE Tunnels

Configuring Ethernet over GRE (EoGRE) on the mobile node.

```
! configure the topology
mobile-node1(config-if)# interface GigabitEthernet0/1
mobile-node1(config-if)# ip address 10.21.1.1 255.255.255.0
mobile-node1(config-if)# no shutdown
mobile-node1(config-if)# exit
mobile-node1(config)# ip route 10.0.0.1 255.255.255.255 10.21.1.2

! Configure the interface used as the source of the tunnel
mobile-node1(config)# interface Loopback0
mobile-node1(config-if)# ip address 10.40.0.1 255.255.255.0
mobile-node1(config-if)# ipv6 address 2001:db8:2:40::1/64
mobile-node1(config-if)# no shutdown
```

```
! Configure the Ethernet over GRE IPv4 Tunnel
mobile-node1(config-if)# interface Tunnel1
mobile-node1(config-if)# mac-address 0000.0000.0001
mobile-node1(config-if)# ip dhcp client client-id ascii MN1@cisco.com
mobile-node1(config-if)# ip address dhcp
mobile-node1(config-if)# no ip redirects
mobile-node1(config-if)# no ip route-cache
mobile-node1(config-if)# tunnel source Loopback0
mobile-node1(config-if)# tunnel mode ethernet gre ipv4
mobile-node1(config-if)# tunnel key 1
mobile-node1(config-if)# tunnel vlan 10, 20
mobile-node1(config-if)# no shutdown
mobile-node1(config-if)# exit

Configuring Ethernet over GRE tunnel on the MAG

! Configure the topology
MAG(config)# interface FastEthernet1/1/5
MAG(config-if)# ip address 10.21.1.2 255.255.255.0
MAG(config-if)# ipv6 address 2001:db8:2:21::2/64
MAG(config-if)# no shutdown
MAG(config)# ip route 10.40.0.1 255.255.255.255 10.21.1.1

! Configure the interface used as source of the tunnel
MAG(config-if)# interface Loopback0
MAG(config-if)# ip address 10.0.0.1 255.255.255.0
MAG(config-if)# no shutdown

! configure the policy map
MAG(config)# policy-map tunnel-qos-policy
MAG(config-pmap)# class class-default
MAG(config-pmap-c)# police rate 200000 bps
MAG(config-pmap-c)# exit

! Configure the Ethernet over GRE IPv4 Tunnel
MAG(config)# interface Tunnel1
MAG(config-if)# ip address 10.11.1.1 255.255.255.0
MAG(config-if)# tunnel mode ethernet gre ipv4
MAG(config-if)# tunnel source Loopback0

! Configure a static GRE and VLAN ID for the tunnel
MAG(config-if)# tunnel key 1
MAG(config-if)# tunnel vlan 10, 20

!Associate the QoS policy to the tunnel interface
MAG(config-if)# tunnel endpoint service-policy output tunnel-qos-policy

! Enable ISG on the tunnel
MAG(config-if)# ip subscriber l2-connected
MAG(config-subscriber)# initiator unclassified mac-address
MAG(config-subscriber)# initiator dhcp
MAG(config-subscriber)# exit
```

Additional References for QoS on Ethernet over GRE Tunnels

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Ethernet over GRE Tunnels	*Cisco IOS Interface and Hardware Component Configuration Guide*
Tunnel commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Interface and Hardware Component Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for QoS on Ethernet over GRE Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 327: Feature Information for QoS on Ethernet over GRE Tunnels

Feature Name	Releases	Feature Information
QoS on Ethernet over GRE Tunnels	Cisco IOS XE 3.13S	The QoS on Ethernet over GRE (EoGRE) Tunnels feature enables service providers to configure a common QoS policy for all endpoints. This feature supports dual high availability for a route processor. The following command was introduced by this feature: **tunnel endpoint service-policy output**.

CHAPTER 270

VRF-Aware IPv6 Rapid Deployment Tunnel

Virtual Routing and Forwarding - aware tunnels are used to connect customer networks separated by untrusted core networks or core networks with different infrastructures (IPv4 or IPv6). The VRF-Aware IPv6 Rapid Deployment Tunnel feature extends Virtual Routing and Forwarding (VRF) awareness to IPv6 rapid deployment tunnels.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for the VRF-Aware IPv6 Rapid Deployment Tunnel

The VRF- Aware IPv6 Rapid Deployment Tunnel feature has the following restrictions:

- The incoming physical interface, and the tunnel interface should have the same VRF instance defined.
- The tunnel transport VRF and the egress physical interface, through which the traffic leaves should have the same VRF instance defined.
- For IPv6 rapid deployment Customer Edge (CE) router configuration, the tunnel source and the Border Relay (BR) router address should have the same VRF instance defined as the physical interface through which the traffic flows.

Information About the VRF-Aware IPv6 Rapid Deployment Tunnel

The IPv6 Rapid Deployment Tunnel feature allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4. Currently, the IPv6 Rapid Deployment Tunnel feature does not support VRF. Therefore, the forwarding table look up tasks for locating IPv6 overlay addresses and IPv4 transport addresses are performed in the global routing table. The VRF-Aware IPv6 Rapid Deployment Tunnel feature extends the IPv6 rapid deployment tunneling support for IPv6 overlay addresses and IPv4 transport addresses in VRF.

The following scenarios are supported for VRF-Aware IPv6 Rapid Deployment Tunnel feature:

- The IPv6 rapid deployment tunnel is in the VRF and both IPv6 overlay address and the IPv4 transport address are in VRF.
- IPv6 rapid deployment tunnel and the IPv4 address are in VRF. The incoming global routing table IPv6 traffic selects the correct VRF, based on the IPv6 rapid deployment domain.

The following figure explains the topology and sample configurations for the VRF Aware IPv6 Rapid Deployment Tunnel feature where both the IPv4 addresses and IPv6 addresses are in VRF.

Figure 226: Topology of the VRF-Aware IPv6 Rapid Deployment Tunnel

How to Configure the VRF-Aware IPv6 Rapid Deployment Tunnel

Complete the steps in the following procedure to configure the VRF-Aware IPv6 Rapid Deployment Tunnel feature when both the IPv6 and IPv4 addresses are in VRF. You should perform these steps on the CE router and BR router unless specifically mentioned otherwise in the following procedure.

Configuring the VRF-Aware IPv6 Rapid Deployment Tunnel

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **vrf definition** *vrf-name1*
5. **rd** {*ASN:nn* | *IP address: nn*}
6. **route-target** [**import | export | both**] {*ASN:nn* |*IP address: nn*}
7. **address-family ipv6**
8. **exit**
9. **address-family ipv4**
10. **exit**
11. **exit**
12. **vrf definition** *vrf-name2*
13. **rd** {*ASN:nn* | *IP address: nn*}
14. **route-target** [**import | export | both**] { *ASN:nn* | *IP address: nn*}
15. **address-family ipv4**
16. **exit**

17. **exit**
18. **interface gigabitethernet** *slot / port*
19. **vrf forwarding** *vrf-name1*
20. **ipv6 address** {*ipv6-address prefix-length prefix-name sub-bits prefix-length*}
21. **exit**
22. **interface gigabitethernet** *slot* / *port*
23. **vrf forwarding** *vrf-name2*
24. **ip address** *ip-address mask*
25. **exit**
26. **interface loopback** *interface-number*
27. **vrf forwarding** *vrf-name2*
28. **ip address** *ip-address*
29. **exit**
30. **interface tunnel** *tunnel-number*
31. **vrf forwarding** *vrf-name1*
32. **ipv6 address** {*ipv6-address prefix-length prefix-name sub-bits prefix-length*}
33. **tunnel source**{ *ip-address*| *interface-type interface-number*}
34. **tunnel mode ipv6ip** [**6rd**| **6to4** | **auto-tunnel** | **isatap**]
35. **tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length*}
36. **tunnel 6rd prefix**{*ipv6-prefix*/ *prefix-length*}
37. **tunnel 6rd br** *ipv4-address*
38. **tunnel vrf** *vrf-name2*
39. **exit**
40. **ipv6 route vrf** *vrf-name1* {*ipv6-prefix* / *prefix-length*} **tunnel** *tunnel-number*
41. **ipv6 route vrf** *vrf-name1*{*ipv6-prefix*/ *prefix-length*} **tunnel** *tunnel-number ipv6-address*
42. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables the privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters the global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the task of forwarding IPv6 unicast datagrams.
Step 4	**vrf definition** *vrf-name1* **Example:** `Router(config)# vrf definition VRF_RED`	Configures a VRF instance and enters the VRF configuration mode.

	Command or Action	Purpose
Step 5	**rd** {*ASN:nn* | *IP address: nn*} **Example:** `Router(config-vrf)# rd 1:1`	Specifies a route distinguisher. • *ASN:nn* — Specifies an autonomous system number and an arbitrary number. • *IP address: nn* — Specifies an IP address and an arbitrary number.
Step 6	**route-target** [**import | export | both**] {*ASN:nn* |*IP address: nn*} **Example:** `Router(config-vrf)# route-target import 1:1`	Creates a route target extended community for a VRF instance. Route target extended community attributes are used to identify a set of sites and VRF instances that can receive routes with a configured route target. • **import** — Imports routing information from the target VPN extended community. • **export** — Exports routing information to the target VPN extended community. • **both** — Imports both import and export routing information to the target VPN extended community • *ASN:nn* — Specifies an autonomous system number and an arbitrary number. • *IP address: nn* — Specifies an IP address and an arbitrary number.
Step 7	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Selects IPv6 as address family type for a VRF table and enters VRF address family configuration mode. Configures separate route-target policies for IPv6.
Step 8	**exit** **Example:** `Router(config-vrf-af)# exit`	Exits the address family configuration mode.
Step 9	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Selects IPv4 as address family type for a VRF table and enters VRF address family configuration mode. Configures the separate route-target policies for IPv4.
Step 10	**exit** **Example:** `Router(config-vrf-af)# exit`	Exits the address family configuration mode.
Step 11	**exit** **Example:** `Router(config-vrf)# exit`	Exits the VRF configuration mode.

	Command or Action	Purpose
Step 12	**vrf definition** *vrf-name2* **Example:** `Router(config)# vrf definition VRF_GREEN`	Configures a VRF instance and enters the VRF configuration mode.
Step 13	**rd** {*ASN:nn* | *IP address: nn*} **Example:** `Router(config-vrf)# rd 1:1`	Specifies a route distinguisher.
Step 14	**route-target** [**import | export | both**] { *ASN:nn* | *IP address: nn*} **Example:** `Router(config-vrf)# route-target import 1:1`	Creates a route-target extended community for a VRF instance. Route-target extended community attributes are used to identify a set of sites and VRF instances that can receive routes with a configured route target • **import** — Imports routing information from the target VPN extended community. • **export** — Exports routing information to the target VPN extended community. • **both** — Imports and exports routing information to the target VPN extended community and from the target VPN extended community. • *ASN:nn* — Specifies an autonomous system number and an arbitrary number. • *IP address: nn* — Specifies an IP address and an arbitrary number.
Step 15	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Selects IPv4 as address family for a VRF table and enters the VRF address family configuration mode. Configures separate route-target policies for IPv4.
Step 16	**exit** **Example:** `Router(config-vrf-af)# exit`	Exits the address family configuration mode.
Step 17	**exit** **Example:** `Router(config-vrf)# exit`	Exits the VRF configuration mode.
Step 18	**interface gigabitethernet** *slot / port* **Example:** `Router(config)# interface gigabitethernet 3/1`	Enters the interface configuration mode and specifies the Gigabit interface to configure.
Step 19	**vrf forwarding** *vrf-name1* **Example:** `Router(config-if)# vrf forwarding VRF_RED`	Associates a VRF instance with an interface or a subinterface.

	Command or Action	Purpose
Step 20	**ipv6 address** {*ipv6-address prefix-length prefix-name sub-bits prefix-length*} **Example:** `Router(config-if)# ipv6 address 1::2/64`	Specifies the IPv6 address assigned to the interface, and enables IPv6 processing on the interface.
Step 21	**exit** **Example:** `Router(config-if)# exit`	Exits the interface configuration mode.
Step 22	**interface gigabitethernet** *slot* / *port* **Example:** `Router(config)# interface gigabitethernet 4/5`	Enters the interface configuration mode and specifies the Gigabit interface to configure.
Step 23	**vrf forwarding** *vrf-name2* **Example:** `Router(config-if)# vrf forwarding VRF_GREEN`	Associates a VRF instance with an interface or a subinterface.
Step 24	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 17.1.1.1 255.255.255.0`	Assigns an IP address and subnet mask to the interface.
Step 25	**exit** **Example:** `exit`	Exits the interface configuration mode.
Step 26	**interface loopback** *interface-number* **Example:** `Router(config)# interface Loopback 100`	Enters the interface configuration mode and specifies the new loopback interface.
Step 27	**vrf forwarding** *vrf-name2* **Example:** `Router(config-if)# vrf forwarding VRF_GREEN`	Associates a VRF instance with an interface or a subinterface.
Step 28	**ip address** *ip-address* **Example:** `Router(config-if)# ip address 60.1.1.1 255.255.255.0`	Assigns an IP address and subnet mask to the loopback interface.
Step 29	**exit** **Example:** `Router(config-if)# exit`	Exits the interface configuration mode.

	Command or Action	Purpose
Step 30	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 10`	Specifies a tunnel interface and enters the interface configuration mode.
Step 31	**vrf forwarding** *vrf-name1* **Example:** `Router(config-if)# vrf forwarding VRF_RED`	Associates a VRF instance with an interface or a subinterface.
Step 32	**ipv6 address** {*ipv6-address prefix-length prefix-name sub-bits prefix-length*} **Example:** `Router(config-if)# ipv6 address 2001:A000:100::1/128`	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface.
Step 33	**tunnel source**{ *ip-address*| *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source loopback 100`	Specifies the source interface type and number for the tunnel interface.
Step 34	**tunnel mode ipv6ip** [**6rd**| **6to4** | **auto-tunnel** | **isatap**] **Example:** `Router(config-if)# tunnel mode ipv6ip 6rd`	Configures a static IPv6 tunnel interface.
Step 35	**tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length*} **Example:** `Router(config-if)# tunnel 6rd ipv4 prefix-len 16 suffix-len 8`	Specifies the prefix and suffix length of the IPv4 transport address that is common to all the 6rd tunnels.
Step 36	**tunnel 6rd prefix**{*ipv6-prefix*/ *prefix-length*} **Example:** `Router(config-if)# tunnel 6rd prefix 2001:A000::/32`	Specifies the common IPv6 prefix on IPv6 6rd tunnels.
Step 37	**tunnel 6rd br** *ipv4-address* **Example:** `Router(config-if)# tunnel 6rd br 60.1.2.1`	Bypasses security checks on a 6rd CE router. • *ipv4-address* — IPv4 address of the border relay (BR) router. **Note** Perform this step only on a CE router, not on a BR router.
Step 38	**tunnel vrf** *vrf-name2* **Example:** `Router(config-if)# tunnel vrf  VRF_GREEN`	Configures a VRF instance with a specific tunnel destination, interface, or a subinterface. **Note** This command specifies the VRF instance used for the tunnel IPv4 transport address lookup.

	Command or Action	Purpose
Step 39	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 40	**ipv6 route vrf** *vrf-name1* {*ipv6-prefix* / *prefix-length*} **tunnel** *tunnel-number* **Example:** `Router(config)# ipv6 route vrf VRF_RED 2001:A000::/32 Tunnel10`	Establishes static routes. • *ipv6-prefix* — Specifies the IPv6 network that is the destination of the static route. Can also be a host name when static host routes are configured. • *ipv6-address* — The IPv6 address of the next hop that can be used to reach the specified network..
Step 41	**ipv6 route vrf** *vrf-name1*{*ipv6-prefix*/ *prefix-length*} **tunnel** *tunnel-number ipv6-address* **Example:** `Router(config)# ipv6 route vrf VRF_RED 9000:1000::/64 Tunnel10 2001:A000:200::1`	Establishes static routes. • *ipv6-prefix* — Specifies the IPv6 network that is the destination of the static route. Can also be a host name when static host routes are configured. • *prefix-length* — Specifies the length of the IPv6 prefix.
Step 42	**end** **Example:** `Router(config)# end`	Ends the current configuration session.

Example: Configuring VRF- Aware IPv6 Rapid Deployment Tunnel

The following example shows how to configure the VRF-Aware IPv6 Rapid Deployment Tunnel on a CE router:

```
Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# mls ipv6 vrf
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# vrf definition VRF_GREEN
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# interface gigabitethernet 3/1
Router(config-if)# vrf forwarding VRF_RED
```

```
Router(config-if)# ipv6 address 1::2/64
Router(config-if)# exit
Router(config)# interface gigabitethernet 4/5
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 17.1.1.1 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface Loopback 100
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 60.1.1.1 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface tunnel 10
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 2001:A000:100::1/128
Router(config-if)# mls 6rd reserve interface GigabitEthernet4/5
Router(config-if)# tunnel source loopback 100
Router(config-if)# tunnel mode ipv6ip 6rd
Router(config-if)# tunnel 6rd ipv4 prefix-len 16 suffix-len 8
Router(config-if)# tunnel 6rd prefix 2001:A000::/32
Router(config-if)# tunnel 6rd br 60.1.2.1
Router(config-if)# tunnel vrf  VRF_GREEN
Router(config-if)# exit
Router(config)# ipv6 route vrf VRF_RED 2001:A000::/32 Tunnel10
Router(config)# ipv6 route vrf VRF_RED 9000:1000::/64 Tunnel10 2001:A000:200::1
Router(config)# end
```

The following example shows how to configure the VRF-Aware IPv6 Rapid Deployment Tunnel on a BR router:

```
Router# enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# vrf definition VRF_RED
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# vrf definition VRF_GREEN
Router(config-vrf)# rd 1:1
Router(config-vrf)# route-target export 1:1
Router(config-vrf)# route-target import 1:1
Router(config-vrf)# address-family ipv4
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# interface gigabitethernet 5/1
Router(config-if)# vrf forwarding VRF_RED
Router(config-if)# ipv6 address 9000:1000::/64
Router(config-if)# exit
Router(config)# interface gigabitethernet 4/1
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 17.1.1.2 255.255.255.0
Router(config-if)# ip ospf 2 area 0
Router(config-if)# exit
Router(config)# interface Loopback 100
Router(config-if)# vrf forwarding VRF_GREEN
Router(config-if)# ip address 60.1.2.1 255.255.255.0
Router(config-if)# exit
Router(config)# interface tunnel 10
Router(config-if)# vrf forwarding VRF_RED
```

```
Router(config-if)# ipv6 address 2001:A000:100::1/128
Router(config-if)# tunnel source loopback 100
Router(config-if)# tunnel mode ipv6ip 6rd
Router(config-if)# tunnel 6rd ipv4 prefix-len 16 suffix-len 8
Router(config-if)# tunnel 6rd prefix 2001:A000::/32
Router(config-if)# tunnel vrf  VRF_GREEN
Router(config-if)# exit
Router(config)# ipv6 route vrf VRF_RED 2001:A000::/32 Tunnel10
Router(config)# end
```

Feature Information for the VRF-Aware IPv6 Rapid Deployment Tunnel

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 328: Feature Information for the VRF-Aware IPv6 Rapid Deployment Tunnel

Feature Name	Releases	Feature Information
VRF-Aware IPv6 Rapid Deployment Tunnel	Cisco IOS XE Release 3.10S	The IPv6 Rapid Deployment Tunnel feature allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4. The VRF-Aware IPv6 Rapid Deployment Tunnel feature extends VRF awareness to IPv6 rapid deployment tunnels.

CHAPTER 271

IP Tunnel - GRE Key Entropy Support

The IP Tunnel - GRE Key Entropy Support feature enables load balancing of tunnel packets in the Generic Routing Encapsulation (GRE) mode of a core network.

Prerequisites for IP Tunnel - GRE Key Entropy Support

- You can enable tunnel entropy calculation only on Generic Routing Encapsulation (GRE) mode of the tunnel interface.
- You must configure the tunnel key value before you enable tunnel entropy calculation.

Restrictions for IP Tunnel - GRE Key Entropy Support

- You must not configure a tunnel key with a value that is more than 24 bits. The configuration of tunnel entropy calculation fails if the tunnel key value is more than 24 bits.
- You cannot disable tunnel entropy calculation unless you remove the configured tunnel key.

Information About IP Tunnel - GRE Key Entropy Support

IP Tunnel - GRE Key Entropy Support Overview

The IP Tunnel - GRE Key Entropy Support feature enables load balancing of tunnel packets in the Generic Routing Encapsulation (GRE) mode of a core network. You can configure the tunnel entropy calculation feature only on the GRE mode of the tunnel interface.

The characteristics of a tunnel entropy label are:

- You cannot use entropy labels for packet forwarding.
- You cannot use entropy labels for signaling.
- You can only use the entropy label to improve load balancing on a network.

In order to configure tunnel entropy calculation using the **tunnel entropy** command, you must first configure a tunnel key using the **tunnel key** command in interface configuration mode. The tunnel key has a maximum size of 32 bits. If you configure tunnel entropy calculation, 24 bits are reserved for the GRE key and 8 bits for entropy.

Note If you configure a GRE tunnel key of 32 bits, you cannot configure tunnel entropy calculation. You must remove the tunnel key and then configure a key of the size of 24 bits or less. To disable an already configured GRE tunnel entropy, remove the GRE tunnel key value first.

Entropy bits are calculated by 6 tuples, which are virtual routing and forwarding (VRF) ID, source IP address, destination IP address, source port, destination port, and protocols of the private IPv4/IPv6 packets in a network.

How To Configure IP Tunnel - GRE Key Entropy Support

Configuring IP Tunnel - GRE Key Entropy Support

Perform this task to configure GRE tunnel entropy calculation:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ipv4-addr* | *ipv6-addr* | *interface-type interface-number* | **dynamic**}
5. **tunnel destination** {*ipv4-addr* | *ipv6-addr* | *hostname* | **dynamic**}
6. **tunnel mode gre ip**
7. **tunnel key** *key-number*
8. **tunnel entropy**
9. **end**
10. **show interfaces** *interface-type interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 21`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**tunnel source** {*ipv4-addr* | *ipv6-addr* | *interface-type interface-number* | **dynamic**} **Example:** `Device(config-if)# tunnel source 10.1.1.1`	Specifies the source IP address for a tunnel interface.
Step 5	**tunnel destination** {*ipv4-addr* | *ipv6-addr* | *hostname* | **dynamic**} **Example:** `Device(config-if)# tunnel destination 172.168.2.1`	Specifies the destination IP address for a tunnel interface.
Step 6	**tunnel mode gre ip** **Example:** `Device(config-if)# tunnel mode gre ip`	Configures the encapsulation mode for a tunnel interface.
Step 7	**tunnel key** *key-number* **Example:** `Device(config-if)# tunnel key 4683`	Enables an ID key for a tunnel interface.
Step 8	**tunnel entropy** **Example:** `Device(config-if)# tunnel entropy`	Achieves load balancing of tunnel packets in a network.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 10	**show interfaces** *interface-type interface-number* **Example:** `Device# show interfaces tunnel 21`	Displays statistics for all interfaces configured on a device or access server.

Configuration Examples for IP Tunnel - GRE Key Entropy Support

Examples: Configuring IP Tunnel - GRE Key Entropy Support

The following example shows how to configure tunnel entropy calculation for GRE mode of the tunnel interface:

```
Device> enable
Device# configure terminal
Device(config)# interface tunnel 21
Device(config-if)# tunnel source 10.1.1.1
Device(config-if)# tunnel destination 172.168.2.1
Device(config-if)# tunnel mode gre ip
Device(config-if)# tunnel key 4683
Device(config-if)# tunnel entropy
Device(config-if)# end
```

The following is sample output from the **show interfaces tunnel** command, which displays that tunnel entropy calculation is enabled with a 24-bit key:

```
Device# show interfaces tunnel 21

Tunnel21 is up, line protocol is up
Hardware is Tunnel
MTU 17864 bytes, BW 100 Kbit/sec, DLY 50000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.1.1.1, destination 172.168.2.1
Tunnel protocol/transport GRE/IP
Key 0x124B, sequencing disabled
Checksumming of packets disabled
Tunnel Entropy Calculation Enabled (24-bit Key)
Tunnel TTL 255, Fast tunneling enabled
Tunnel transport MTU 1472 bytes
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Last input never, output never, output hang never
Last clearing of "show interface" counters 00:03:07
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 output buffer failures, 0 output buffers swapped out
```

Additional References for IP Tunnel - GRE Key Entropy Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Tunnel commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS Interface and Hardware Component Command Reference*
Cisco IOS XE Interface and Hardware Component configuration modules	*Cisco IOS XE Interface and Hardware Component Configuration Guide*

Standards and RFCs

RFC	Title
RFC6790	The Use of Entropy Labels in MPLS Forwarding

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	*http://www.cisco.com/support*

Feature Information for IP Tunnel - GRE Key Entropy Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 329: Feature Information for IP Tunnel - GRE Key Entropy Support

Feature Name	Releases	Feature Information
IP Tunnel - GRE Key Entropy Support	Cisco IOS XE Release 3.11S	The IP Tunnel - GRE Key Entropy Support feature enables load balancing of tunnel packets in the Generic Routing Encapsulation (GRE) mode of a core network. The following commands were introduced or modified: **tunnel entropy** **tunnel key** **show interfaces**

PART X

Multitopology Routing

CHAPTER 272

IS-IS Support for MTR

The IS-IS Support for MTR feature provides Intermediate System-to-Intermediate System (IS-IS) support for multiple logical topologies over a single physical network. This module describes how to configure IS-IS for Multitopology Routing (MTR) for both unicast and multicast topologies.

Prerequisites for IS-IS Support for MTR

- Be familiar with the concepts in the “Routing Protocol Support for MTR” section.
- Configure and activate a global topology configuration.
- You must configure a multicast topology before activating the Intermediate System-to-Intermediate System (IS-IS) protocol in the multicast topology. For details, see the “MTR support for Multicast” feature module.
- Activate a Multitopology Routing (MTR) topology on an IS-IS device.
- Configure the MTR topology to globally configure all interfaces by using the **all-interfaces** address family topology configuration command, or configure the IS-IS topology in interface configuration mode to configure only IS-IS interfaces. The order in which you perform the two tasks does not matter.

Restrictions for IS-IS Support for MTR

Only the IPv4 address family (multicast and unicast) and IPv6 address family unicast are supported. For information about configuring Multitopology IS-IS for IPv6, see the *IS-IS Configuration Guide.*

../topics/Information About IS-IS Support for MTR

Routing Protocol Support for MTR

You must enable IP routing on the device for Multitopology Routing (MTR) to operate. MTR supports static and dynamic routing in Cisco software. You can enable dynamic routing per topology to support interdomain and intradomain routing. Route calculation and forwarding are independent for each topology. MTR support is integrated into Cisco software for the following protocols:

- Border Gateway Protocol (BGP)
- Integrated Intermediate System-to-Intermediate System (IS-IS)

You apply the per-topology configuration in router address family configuration mode of the global routing process (router configuration mode). The address family and subaddress family are specified when the device enters address family configuration mode. You specify the topology name and topology ID by entering the **topology** command in address family configuration mode.

You configure each topology with a unique topology ID under the routing protocol. The topology ID is used to identify and group Network Layer Reachability Information (NLRI) for each topology in updates for a given protocol. In OSPF, EIGRP, and IS-IS, you enter the topology ID during the first configuration of the **topology** command for a class-specific topology. In BGP, you configure the topology ID by entering the **bgp tid** command under the topology configuration.

You can configure class-specific topologies with different metrics than the base topology. Interface metrics configured on the base topology can be inherited by the class-specific topology. Inheritance occurs if no explicit inheritance metric is configured in the class-specific topology.

You configure BGP support only in router configuration mode. You configure Interior Gateway Protocol (IGP) support in router configuration mode and in interface configuration mode.

By default, interfaces are not included in nonbase topologies. For routing protocol support for EIGRP, IS-IS, and OSPF, you must explicitly configure a nonbase topology on an interface. You can override the default behavior by using the **all-interfaces** command in address family topology configuration mode. The **all-interfaces** command causes the nonbase topology to be configured on all interfaces of the device that are part of the default address space or the virtual routing and forwarding (VRF) instance in which the topology is configured.

Interface Configuration Support for MTR

The configuration of a Multitopology Routing (MTR) topology in interface configuration mode allows you to enable or disable MTR on a per-interface basis. By default, a class-specific topology does not include any interfaces.

You can include or exclude individual interfaces by configuring the **topology** interface configuration command. You specify the address family and the topology (base or class-specific) when entering this command. The subaddress family can be specified. If no subaddress family is specified, the unicast subaddress family is used by default.

You can include globally all interfaces on a device in a topology by entering the **all-interfaces** command in routing topology configuration mode. Per-interface topology configuration applied with the **topology** command overrides global interface configuration.

The interface configuration support for MTR has these characteristics:

- Per-interface routing configuration: Interior Gateway Protocol (IGP) routing and metric configurations can be applied in interface topology configuration mode. Per-interface metrics and routing behaviors can be configured for each IGP.
- Open Shortest Path First (OSPF) interface topology configuration: Interface mode OSPF configurations for a class-specific topology are applied in interface topology configuration mode. In this mode, you can configure an interface cost or disable OSPF routing without removing the interface from the global topology configuration.
- Enhanced Interior Gateway Routing Protocol (EIGRP) interface topology configuration: Interface mode EIGRP configurations for a class-specific topology are applied in interface topology configuration mode. In this mode, you can configure various EIGRP features.
- Intermediate System-to-Intermediate System (IS-IS) interface topology configuration: Interface mode IS-IS configurations for a class-specific topology are applied in interface topology configuration mode. In this mode, you can configure an interface cost or disable IS-IS routing without removing the interface from the global topology configuration.

../topics/How to Configure IS-IS Support for MTR

Activating an MTR Topology by Using IS-IS

Note Only Multitopology Routing (MTR) commands are shown in this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **net** *network-entity-title*
5. **metric-style wide** [**transition**] [**level-1** | **level-2** | **level-1-2**]
6. **address-family ipv4** [**multicast** | **unicast**]
7. **topology** *topology-name* **tid** *number*
8. **end**
9. **show isis neighbors detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Device(config)# router isis`	Enables the Intermediate System-to-Intermediate System (IS-IS) routing protocol and optionally specifies an IS-IS process. • Enters router configuration mode.
Step 4	**net** *network-entity-title* **Example:** `Device(config-router)# net 31.3131.3131.3131.00`	Configures an IS-IS network entity title (NET) for a Connectionless Network Service (CLNS) routing process.
Step 5	**metric-style wide** [**transition**] [**level-1** | **level-2** | **level-1-2**] **Example:** `Device(config-router)# metric-style wide`	Globally changes the metric value for all IS-IS interfaces. **Note** Wide style metrics are required for prefix tagging.
Step 6	**address-family ipv4** [**multicast** | **unicast**] **Example:** `Device(config-router)# address-family ipv4`	Enters router address family configuration mode.
Step 7	**topology** *topology-name* **tid** *number* **Example:** `Device(config-router-af)# topology DATA tid 100`	Configures IS-IS support for the topology and assigns a Topology Identifier (TID) number for each topology. • In this example, IS-IS support for the DATA topology is configured.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits router address family configuration mode and returns to privileged EXEC mode.
Step 9	**show isis neighbors detail** **Example:** `Device# show isis neighbors detail`	(Optional) Displays information about IS-IS neighbors, including MTR information for the TID values for the device and its IS-IS neighbors.

What to Do Next

If a Border Gateway Protocol (BGP) topology configuration is required, see the “BGP Support for MTR” feature module.

Activating an MTR Topology in Interface Configuration Mode by Using IS-IS

Before you begin

Define a topology globally before performing the per-interface topology configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **ip router isis** [*area-tag*]
6. **topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]}
7. **isis topology disable**
8. **topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]}
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 2/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 192.168.7.17 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip router isis** [*area-tag*] **Example:** `Device(config-if)# ip router isis`	Configures an Intermediate System-to-Intermediate System (IS-IS) routing process for IP on an interface and attaches an area designator to the routing process. **Note** If a tag is not specified, a null tag is assumed and the process is referenced with a null tag.

	Command or Action	Purpose
Step 6	**topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]} **Example:** `Device(config-if)# topology ipv4 DATA`	Configures a Multitopology Routing (MTR) topology instance on an interface and enters interface topology configuration mode. **Note** In this example, the topology instance DATA is configured for an MTR network that has a global topology named DATA.
Step 7	**isis topology disable** **Example:** `Device(config-if-topology)# isis topology disable`	(Optional) Prevents an IS-IS process from advertising the interface as part of the topology. **Note** In this example, the topology instance DATA will not advertise the interface as part of the topology.
Step 8	**topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]} **Example:** `Device(config-if-topology)# topology ipv4 VOICE`	Configures an MTR topology instance on an interface. **Note** In this example, the topology instance VOICE is configured for an MTR network that has a global topology named VOICE.
Step 9	**end** **Example:** `Device(config-if-topology)# end`	Exits interface topology configuration mode and returns to privileged EXEC mode.

Monitoring Interface and Topology IP Traffic Statistics for MTR

Use any of the following commands in any order to monitor interface and topology IP traffic statistics for Multitopology Routing (MTR).

SUMMARY STEPS

1. **enable**
2. **show ip interface** [*type number*] [**topology** {*name* | **all** | **base**}] [**stats**]
3. **show ip traffic** [**topology** {*name* | **all** | **base**}]
4. **clear ip interface** *type number* [**topology** {*name* | **all** | **base**}] [**stats**]
5. **clear ip traffic** [**topology** {*name* | **all** | **base**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip interface** [*type number*] [**topology** {*name* | **all** | **base**}] [**stats**] **Example:** `Device# show ip interface FastEthernet 1/10 stats`	(Optional) Displays IP traffic statistics for all interfaces or statistics related to the specified interface. • If you specify an interface type and number, information for that specific interface is displayed. If you specify no optional arguments, information for all the interfaces is displayed. • If the **topology** *name* keyword and argument are used, statistics are limited to the IP traffic for that specific topology. • The **base** keyword displays the IPv4 unicast base topology.
Step 3	**show ip traffic** [**topology** {*name* | **all** | **base**}] **Example:** `Device# show ip traffic topology VOICE`	(Optional) Displays global IP traffic statistics (an aggregation of all the topologies when MTR is enabled) or statistics related to a particular topology. • The **base** keyword is reserved for the IPv4 unicast base topology.
Step 4	**clear ip interface** *type number* [**topology** {*name* | **all** | **base**}] [**stats**] **Example:** `Device# clear ip interface FastEthernet 1/10 topology all`	(Optional) Resets interface-level IP traffic statistics. • If the **topology** keyword and a related keyword are not used, only the interface-level aggregate statistics are reset. • If all topologies need to be reset, use the **all** keyword as the topology name.
Step 5	**clear ip traffic** [**topology** {*name* | **all** | **base**}] **Example:** `Device# clear ip traffic topology all`	(Optional) Resets IP traffic statistics. • If no topology name is specified, global statistics are cleared.

../topics/Configuration Examples for IS-IS Support for MTR

Example: Activating an MTR Topology by Using IS-IS

The following example shows how to configure both the Multitopology Routing (MTR) topologies DATA and VIDEO and Intermediate System-to-Intermediate System (IS-IS) support for MTR. The DATA and VIDEO topologies are enabled on three IS-IS neighbors in a network.

Device 1

```
global-address-family ipv4
 topology DATA
```

```
 topology VOICE
 end
interface Ethernet 0/0
 ip address 192.168.128.2 255.255.255.0
 ip router isis
 topology ipv4 DATA
 isis topology disable
 topology ipv4 VOICE
 end
router isis
 net 33.3333.3333.3333.00
 metric-style wide
 address-family ipv4
  topology DATA tid 100
  topology VOICE tid 200
  end
```

Device 2

```
global-address-family ipv4
 topology DATA
 topology VOICE
 all-interfaces
  forward-base
  maximum routes 1000 warning-only
  shutdown
  end
interface Ethernet 0/0
 ip address 192.168.128.1 255.255.255.0
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end
interface Ethernet 1/0
 ip address 192.168.130.1 255.255.255.0
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end
router isis
 net 32.3232.3232.3232.00
 metric-style wide
 address-family ipv4
  topology DATA tid 100
  topology VOICE tid 200
  end
```

Device 3

```
global-address-family ipv4
 topology DATA
  topology VOICE
  all-interfaces
  forward-base
  maximum routes 1000 warning-only
  shutdown
  end
interface Ethernet 1/0
 ip address 192.168.131.1 255.255.255.0
```

```
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end
router isis
 net 31.3131.3131.3131.00
 metric-style wide
 address-family ipv4
  topology DATA tid 100
  topology VOICE tid 200
  end
```

Entering the **show isis neighbors detail** command verifies topology translation with the IS-IS neighbor Device 1:

```
Device# show isis neighbors detail

System Id      Type Interface IP Address      State Holdtime Circuit Id
R1             L2   Et0/0     192.168.128.2   UP    28       R5.01
  Area Address(es): 33
  SNPA: aabb.cc00.1f00
  State Changed: 00:07:05
  LAN Priority: 64
  Format: Phase V
  Remote TID:  100, 200
  Local TID:   100, 200
```

Example: MTR IS-IS Topology in Interface Configuration Mode

The following example shows how to prevent the Intermediate System-to-Intermediate System (IS-IS) process from advertising interface Ethernet 1/0 as part of the DATA topology:

```
interface Ethernet 1/0
 ip address 192.168.130.1 255.255.255.0
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end
```

Additional References

Related Documents

Related Topic	Document Title
Multitopology Routing (MTR) commands	Cisco IOS Multitopology Routing Command Reference
Intermediate System-to-Intermediate System (IS-IS) commands	Cisco IOS IP Routing: IS-IS Command Reference
IS-IS concepts and tasks	*IP Routing: IS-IS Configuration Guide*

Related Topic	Document Title
Configuring a multicast topology	"MTR Support for Multicast" feature module in the *Multitopology Routing Configuration Guide*
Configure Multitopology IS-IS for IPv6	*IP Routing: IS-IS Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IS-IS Support for MTR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 330: Feature Information for IS-IS Support for MTR

Feature Name	Releases	Feature Information
IS-IS Support for MTR	12.2(33)SRB Cisco IOS XE Release 2.5	This feature provides Intermediate System-to-Intermediate System (IS-IS) support for multiple logical topologies over a single physical network. In Cisco IOS XE Release 2.5, support was added for the Cisco ASR 1000 Series Routers. The following commands were introduced or modified: **address-family ipv4**, **isis topology disable**, **show isis neighbors**, **topology**.

CHAPTER 273

MTR in VRF

The MTR in VRF feature extends to IPv4 VRF contexts the Cisco IOS software's capability that allows users to configure one or more non-congruent multicast topologies in global IPv4 routing context. These contexts can be used to forward unicast and multicast traffic over different links in the network, or in the case of non-base topologies to provide a Live-Live multicast service using multiple non-congruent multicast topologies mapped to different (S,G) groups.

Information About MTR in VRF

MTR in VRF Overview

The MTR in VRF feature extends to IPv4 VRF contexts, Cisco IOS software's capability that allows users to configure one or more non-congruent multicast topologies in global IPv4 routing context. These contexts can be used to forward unicast and multicast traffic over different links in the network, or in the case of non-base topologies to provide a Live-Live multicast service using multiple non-congruent multicast topologies mapped to different (S,G) groups.

The Cisco IOS Software allows a set of attributes, primarily used by BGP/MPLS L3VPNs, to be configured on a per-address family basis within a VRF. The MTR in VRF feature allows these attributes to be independently configured for the multicast sub-address families within a VRF address family.

How to Configure VRF in MTR

Configuring MTR in VRF

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **ipv4 multicast multitoplogy**
6. **address-family ipv4**
7. **exit-address-family**
8. **address-family ipv4 multicast**
9. **topology** *topology-instance-name*
10. **all-interfaces**
11. **exit**
12. **exit-address-family**
13. **exit**
14. **interface** *type number*
15. **interface** *type number*
16. **vrf forwarding** *vrf-name*
17. **ip address** *ip-address mask*
18. **ip pim sparse-dense-modeip**
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vd1`	Configures a VRF routing table and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 10:1`	Creates routing and forwarding tables for a VRF.
Step 5	**ipv4 multicast multitoplogy** **Example:** `Device(config-vrf)# ipv4 multicast multitoplogy`	Enables IPv4 multicast support for multi-topology routing (MTR) in a VRF instance.
Step 6	**address-family ipv4** **Example:** `Device(config-vrf)# address-family ipv4`	Specifies the IPv4 address family type and enters address family configuration mode.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode and removes the IPv4 address family.
Step 8	**address-family ipv4 multicast** **Example:** `Device(config-vrf)# address-family ipv4 multicast`	Specifies the IPv4 address family multicast type and enters VRF address family configuration mode.
Step 9	**topology** *topology-instance-name* **Example:** `Device(config-vrf-af)# topology red`	Specifies a topology instance and a name to it and enters VRF address family topology configuration mode.
Step 10	**all-interfaces** **Example:** `Device(config-vrf-af-topology)# all-interfaces`	Configure the topology instance to use all interfaces on the device.
Step 11	**exit** **Example:** `Device(config-vrf-af-topology)# exit`	Exits VRF address-family topology configuration mode and enters VRF address-family configuration mode.
Step 12	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode and removes the IPv4 address family.
Step 13	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 14	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Selects the Ethernet interface and enters the interface configuration mode.
Step 15	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Selects the Ethernet interface and enters the interface configuration mode.
Step 16	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwwarding vrf1`	Associates a VRF instance with the interface.
Step 17	**ip address** *ip-address mask* **Example:**	Sets a primary or secondary IP address for an interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.1.10.1 255.255.255.0`	
Step 18	**ip pim sparse-dense-modeip** **Example:** `Device(config-if)# ip pim sparse-dense-mode`	Enables Protocol Independent Multicast (PIM) on an interface.
Step 19	**end** **Example:** `Device(config-if)# end`	Exits the interface configuration mode and enters privileged EXEC mode.

Configuring Examples for MTR in VRF

Example for MTR in VRF

```
Device> enable
Device# configuration terminal
Device(config)# vrf definition vd1
Device(config-vrf)# rd 10:1
Device(config-vrf)# ipv4 multicast multitopology
Device(config-vrf)# address-family ipv4
Device(config-vrf)# exit-address-family
Device(config-vrf)# address-family ipv4 multicast
Device(config-vrf-af)# topology red
Device(config-vrf-af-topology)# all-interfaces
Device(config-vrf-af-topology)# exit
Device(config-vrf-af)# exit-address-family
Device(config-vrf)# exit
Device(config)# vrf forwarding vrf1
Device(config)# ip address 10.1.10.1 255.255.255.0
Device(config)# ip pim sparse-dense-mode
Device(config)# end
```

Additional References for MTR in VRF

Related Documents

Related Topic	Document Title
Multitopology Routing (MTR) commands	Cisco IOS Multitopology Routing Command Reference
IP multicast commands	Cisco IOS Multicast Command Reference

Related Topic	Document Title
IP multicast concepts and tasks	*IP Multicast Configuration Guide Library*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for MTR in VRF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

CHAPTER 274

Knob for Ping and Traceroute with VRF to Choose Global DNS Server

This feature provides a knob for ping and trace route with VRF to choose global DNS server when no DNS servers are defined in a VRF. This module explains how to configure Knob for Ping and Traceroute with VRF to choose Global DNS Server.

Prerequisites for Knob for Ping and Traceroute with VRF to Choose Global DNS Server

- VRF must be configured.

Information About Knob for Ping and Traceroute with VRF to Choose Global DNS Server

Overview of Knob for Ping and Traceroute with VRF to Choose Global DNS Server

Prior to the Knob for Ping and Traceroute with VRF to choose Global DNS Server feature, ping or traceroute in VRF would look up only in the specified name server to resolve the domain name. If DNS server is specified

in the VRF, the DNS is used to resolve the domain name. If DNS server is not specified in the VRF, the DNS fails to resolve the domain name.

With the implementation of the Knob for Ping and Traceroute with VRF to choose Global DNS Server feature, ping and traceroute uses VRF DNS server (if the server is already configured in a VRF), otherwise global DNS server is used to resolve the domain name. The **ip global-nameserver** command acts as a knob that facilitates the ping and traceroute to use the VRF DNS server or the global DNS server when the server is not configured in a VRF.

../topics/How to Configure Knob for Ping and Traceroute with VRF to Choose Global DNS Server

Configuring a Knob for Ping and Traceroute with VRF to Choose Global DNS Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip global-nameserver**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip global-nameserver** **Example:** `Device(config)# ip global-nameserver`	Configures a knob for ping and traceroute to use VRF DNS server for resolving the domain name.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

../topics/Configuration Examples for Knob for Ping and Traceroute with VRF to Choose Global DNS Server

Example: Knob for Ping and Traceroute with VRF to Choose Global DNS Server

```
Device> enable
Device# configure terminal
Device(config)# ip global-nameserver
Device(config)# exit
```

Additional References for Knob for Ping and Traceroute with VRF to Choose Global DNS Server

Related Documents

Related Topic	Document Title
Multitopology Routing (MTR) commands	Cisco IOS Multitopology Routing Command Reference
MTR in VRF	*Multitopology Routing Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Knob for Ping and Traceroute with VRF to Choose Global DNS Server

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 331: Feature Information for Knob for Ping and Traceroute with VRF to Choose Global DNS Server

Feature Name	Releases	Feature Information
Knob for Ping and Traceroute with VRF to Choose Global DNS Server	Cisco IOS XE Release 3.12S	This feature provides a knob for ping and trace route with VRF to choose global DNS server when no DNS servers are defined in a VRF. The following commands were introduced or modified: **ip global-nameserver**.

PART XI

Performance Routing

CHAPTER 275

Configuring Basic Performance Routing

Performance Routing (PfR) provides additional intelligence to classic routing technologies to track the performance of, or verify the quality of, a path between two devices over a Wide Area Networking (WAN) infrastructure to determine the best egress or ingress path for application traffic.

Cisco Performance Routing complements classic IP routing technologies by adding intelligence to select best paths to meet application performance requirements. The first phase of Performance Routing technology intelligently optimizes application performance over enterprise WANs and to and from the Internet. This technology will evolve to help enable application performance optimization throughout the enterprise network through an end-to-end, performance-aware network.

This document contains an introduction to the basic concepts and tasks required to implement Performance Routing using Cisco IOS XE Software.

Restrictions for Configuring Basic Performance Routing

Only border router functionality is included in the Cisco IOS XE Release 3.1S and 3.2S images; no master controller configuration is available. The master controller that communicates with the Cisco ASR 1000 series router being used as a border router in the Cisco IOS XE Release 3.1S and 3.2S images must be a router running Cisco IOS Release 15.0(1)M, or a later 15.0M release.

Note In Cisco IOS XE Release 3.3S, and later releases, master controller configuration is supported.

Migrating to Cisco-SDWAN from PfR

From Cisco IOS XE 17.4, PfRv3 is not supported. PfRv3 is integrated with IWAN and support for IWAN is discontinued from Cisco IOS XE 17.4 therefore ending the support for PfRv3.

Customers are encouraged to migrate to Cisco's SD-WAN solution, that has Application-aware Routing and a centralized controller to enable SLA-based routing along with performance measurement and monitoring.

The SD-WAN Team has IWAN-migration offerings to ease the transition to Cisco's SD-WAN solution, with particular focus on current IWAN customers, including exclusive commercial offers and technical resources. For more details, reach out to your Cisco representative.

Related Documents:

- https://www.cisco.com/c/dam/en/us/td/docs/routers/sdwan/migration-guide/iwan-to-sdwan-migration-guide.pdf

CHAPTER 276

Performance Routing Version 3

Note From Cisco IOS XE 17.4, PfRv3 is not supported. PfRv3 is integrated with IWAN and support for IWAN is discontinued from Cisco IOS XE 17.4 therefore ending the support for PfRv3. For more information, see the End-of-Sale and End-of-Life Announcement for the Cisco IWAN Release 2.X.X

Performance Routing Version 3 (PfRv3) is the evolution of Performance Routing (PfR). PfRv3 is an intelligent-path control mechanism for improving application delivery and WAN efficiency. It protects critical applications, increases bandwidth utilization, and serves as an integral part of the Cisco Intelligent WAN (IWAN) solution. PfRv3 uses differentiated services code points (DSCP) and application-based policy framework to provide a multi-site aware bandwidth and path control optimization.

Note PfRv3 is not supported beyond Cisco IOS XE Release 17.3

Feature Information for PfRv3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 332: Feature Information for Configuring PfRv3

Feature Name	Releases	Feature Information
PfRv3		Performance Routing v3 (PfRv3) is the evolution of Performance Routing. PfRv3 is an intelligent-path control mechanism for improving application delivery and WAN efficiency. It protects critical applications, increases bandwidth utilization, and serves as an integral part of the Cisco Intelligent WAN (IWAN) solution. The following commands were modified by this feature: **domain default**, **vrf default**, **master**, **source-interface**, **site-prefixes**, **password**, **monitor-interval**, **route-control**, **load-balance**, **enterprise-prefix**, **advanced**, **minimum-mask-length**, **mitigation-mode**, **threshold-variance**, **smart-probes**, **collector**, **class**, **match**, **priority**, **path-preference**, **border**, **domain-path**.

Hardware and Software Support

Cisco Performance Routing Version 3 (PfRv3) supports the following Cisco platforms and software releases:

Device	Cisco IOS Software Release	Hub/Remote Site
Cisco ISR 4000 Series Routers	Cisco IOS XE 3.13 or later	Hub site or remote site
Cisco ASR 1000 Series Routers	Cisco IOS XE 3.13 or later	Hub site
Cisco CSR 1000v Series Routers	Cisco IOS XE 3.14 or later	Hub site (master controller) Branch site (master controller and border router)
Cisco ISR-G2 Series Routers	Cisco IOS 15.5(1)T1 or later Cisco IOS 15.4(3)M1 or later	Remote site

Note PfRv3 is not supported on Cisco Catalyst 8000 Edge Platforms.

Restrictions for Configuring Performance Routing v3

- Asymmetric routing is not supported for application-based policy.
- A new session cannot be established with application-based policy during blackout failure until route converges to backup path. For application-based flows, application ID is not recognized by Network Based Application Recognition (NBAR2) until session gets established and packet exchanges directly. You can configure Differentiated Services Code Point (DSCP) based policy for fast failover with blackout failure.
- PfRv3 does not support High Availability (HA) for both master and border routers. ESP switch over can trigger temporary unreachable event for one to two seconds.
- IPv6 is not supported.
- Network Address Translation (NAT) is not supported.
- Remarking DSCP for traffic flows on WAN interface is not supported.
- On a HUB Master Controller (MC), when a class is configured for matching application within a PFRv3 domain, the list of NBAR application names are limited if there is no active Border Router (BR).

Note Use at atleast one active BR for the MC to display all possible NBAR application names based on the protocol pack installed in BR.

Note PfRv2 is not supported on Cisco IOS 15.6(3)M and Cisco IOS 15.7(3)M or later releases. Cisco IOS XE 16.3.1 has PfRv2 CLIs, but the functionality is not supported.

Migrating to Cisco-SDWAN from PfRv3

From Cisco IOS XE 17.4, PfRv3 is not supported. PfRv3 is integrated with IWAN and support for IWAN is discontinued from Cisco IOS XE 17.4 therefore ending the support for PfRv3.

Customers are encouraged to migrate to Cisco's SD-WAN solution, that has Application-aware Routing and a centralized controller to enable SLA-based routing along with performance measurement and monitoring.

The SD-WAN Team has IWAN-migration offerings to ease the transition to Cisco's SD-WAN solution, with particular focus on current IWAN customers, including exclusive commercial offers and technical resources. For more details, reach out to your Cisco representative.

Related Documents:

- https://www.cisco.com/c/dam/en/us/td/docs/routers/sdwan/migration-guide/iwan-to-sdwan-migration-guide.pdf

Information About PfRv3

Performance Routing v3 Overview

Performance Routing Version 3 (PfRv3) is a one-touch provisioning and multi-site coordination solution that simplifies network provisioning. It enables intelligence of Cisco devices to improve application performance and availability. PfRv3 is an application-based policy driven framework that provides a multi-site aware bandwidth and path control optimization for WAN and cloud-based applications.

PfRv3 monitors network performance and selects best path for each application based on criteria such as reachability, delay, jitter, and loss. It evenly distributes traffic and maintains equivalent link utilization levels and load balances traffic.

It is tightly integrated with existing AVC components such as Performance Monitoring, Quality of Service (QoS), and NBAR2. PfRv3 is useful for enterprise and managed service providers looking for ways to increase their WAN reliability and availability while saving cost.

Benefits of PfRv3

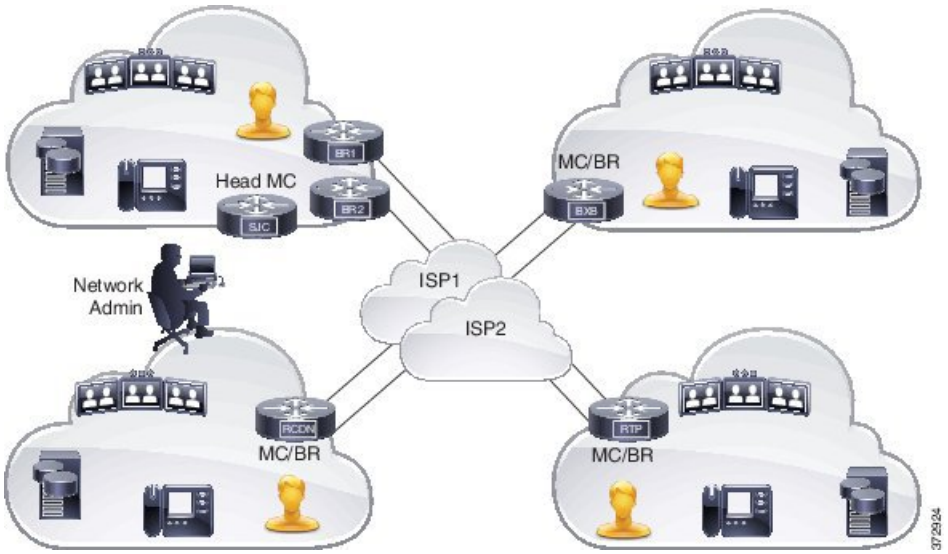
Performance Routing Version 3 provides the following benefits:

- Centralized provisioning — Policies are defined on the hub-master controller and then distributed to all branches. Hence, per-site provisioning is not required in PfRv3.
- Simple provisioning — PfRv3 has simplified policies with pre-existing templates that a user can choose from.
- Enterprise domain — All sites belong to an enterprise domain and are connected with peering.
- Application and DSCP-based policies — Policies are provisioned based on applications. PfRv3 provides application visibility such as bandwidth, performance, and correlation to Quality of Service (QoS) queues by using Unified Monitoring.
- Automatic discovery — PfRv3 site are discovered using peering. Each site peers with the hub site. The WAN interfaces are automatically discovered on the branch sites.
- Scalable passive monitoring — PfRv3 uses Unified Monitor to monitor traffic going into WAN links and traffic coming from the WAN links. It monitors performance metrics based on per DSCP instead of per flow or per prefix basis.
- Smart probing — PfRv3 uses probing mechanism that generates traffic only when there in no traffic. It generates real-time transport protocol traffic, which allows measuring jitter and packet loss using performance monitors.
- Scaling — Smart probing and enhanced passive metrics helps to attain scale up to 2000 branches.
- VRF awareness — Different policies can be configured for different VRFs.

PfRv3 Design Overview

An enterprise organization has a hub and branch site. The hub site consists of master controller and border router.

Figure 227: PfRv3 Design Topology



- In a network, all the policies are created on the hub-master controller. Policies dictate the desired treatment for a set of specified differentiated service code points (DSCPs) or application IDs (such as telepresence, WebEx, and so on) in the network. The policies are percolated to all the master controllers on the network via Service Advertisement Framework (SAF). The policies can be modified by the hub-master controller and the modified policies are sent over the SAF framework so that all the nodes in the network are in sync with the hub-master controller. The hub-master controller collects information about flows handled by border routers. This information is exported to the master controller periodically using the performance monitoring instances (PMI) exporter. A domain can be configured on the central location (Hub) and branches. PfRv3 allows only one domain configuration. Virtual Routing and Forwarding (VRF) and roles are defined on a domain.
- PfRv3 is enabled on the WAN interface of the hub-border routers. The border routers give the flow information to the branch-master controller.
- Every branch has a local-master controller. The master controller can be either co-located with a branch router or a separate router. You must configure both local master and branch border on the same domain. Border devices establishes connection with local-master controller only if both are in the same domain. In a scenario where master and border configurations are on different domain, peering rejects all messages from different peers. Border devices are automatically shut down for five minutes. The connection is established only when the domain conflict is resolved.

  Based on the flow information provided by the hub-border router, the branch-master (local-master) controller applies appropriate controls on the branch router per flow. It ascertains if a flow is operating within the policy limits or out-of-policy. The master-controller to branch-border communication is done via a TCP connection. This connection is used for tasks such as sending configuration and control information from master controller to branch router and flow information from branch router to master controller.

- The branch router is the enforcer, which classifies and measures metrics and sends them to the local-master controller. It is also responsible for path enforcement.

PfRv3 Configuration Components

PfRv3 comprises of the following configuration components:

- Device setup and role — Identifies devices in the network where PfRv3 should be configured and in what role.
- Policy configurations — Identifies the traffic in the network and determines what policies to apply.

Device Setup and Role

There are four different roles a device can play in PfRv3 configuration:

- Hub-master controller — The master controller at the hub site, which can be either a data center or a head quarter. All policies are configured on hub-master controller. It acts as master controller for the site and makes optimization decision.
- Hub-border router — The border controller at the hub site. PfRv3 is enabled on the WAN interfaces of the hub-border routers. You can configure more than one WAN interface on the same device. You can have multiple hub border devices. On the hub-border router, PfRv3 must be configured with the address of the local hub-master controller, path names, and path-ids of the external interfaces. You can use the global routing table (default VRF) or define specific VRFs for the hub-border routers.
- Branch-master controller — The branch-master controller is the master controller at the branch site. There is no policy configuration on this device. It receives policy from the hub-master controller. This device acts as master controller for the branch site and makes optimization decision.
- Branch- border router — The border device at the branch-site. There is no configuration other than enabling of PfRv3 border-master controller on the device. The WAN interface that terminates on the device is detected automatically.

Domain Policies

Domain policies are defined only on the hub-master controller and then sent over peering infrastructure to all the branch-master controllers. Policies can be defined per application or per differentiated service code point (DSCP). You cannot mix and match DSCP and application-based policies in the same class group. Traffic that does not match any of the classification and match statements falls into a default group, which is load balanced (no performance measurement is done).

Note You can either select an existing template for a policy or customize your policies for a domain type.

The following table lists the existing templates for domain type policy:

Pre-defined Template	Threshold Definition
Voice	Priority 1 one-way-delay threshold 150 threshold 150 (msec) Priority 2 packet-loss-rate threshold 1 (%) Priority 2 byte-loss-rate threshold 1 (%) Priority 3 jitter 30 (msec)
Real-time-video	Priority 1 packet-loss-rate threshold 1 (%) Priority 1 byte-loss-rate threshold 1 (%) Priority 2 one-way-delay threshold 150 (msec) Priority 3 jitter 20 (msec)
Low-latency-data	Priority 1 one-way-delay threshold 100 (msec)) Priority 2 byte-loss-rate threshold 5 (%) Priority 2 packet-loss-rate threshold 5 (%)
Bulk-data	Priority 1 one-way-delay threshold 300 (msec) Priority 2 byte-loss-rate threshold 5 (%) Priority 2 packet-loss-rate threshold 5 (%)
Best-effort	Priority 1 one-way-delay threshold 500 (msec) Priority 2 byte-loss-rate threshold 10 (%) Priority 2 packet-loss-rate threshold 10 (%)
Scavenger	Priority 1 one-way-delay threshold 500 (msec) Priority 2 byte-loss-rate threshold 50 (%) Priority 2 packet-loss-rate threshold 50 (%)
Custom	Defines customized user-defined policy values

PfRv3 and Link Group Configuration

PfRv3 allows you to configure the following option for link grouping:

- Allows up to five primary path preferences and four fallback path preferences
- Allows a fallback blackhole configuration
- Allows a fallback routing configuration

During Policy Decision Point (PDP), the exits are first sorted on the available bandwidth and then a second sort algorithm places all primary path preferences in the front of the list followed by fallback preferences. If you have a configuration of primary Internet Service Provider (ISP) 1 and ISP2 and ISP3 as fallback, during policy decision, ISP1 is selected as the primary channel and if ISP2 is equally good it is selected as the fallback. ISP3 is considered only if ISP2 is bad in bandwidth availability.

Routing configuration means that when the traffic is uncontrolled, the routing table takes the responsibility of pushing the flow out of the box.

CHAPTER 277

PfRv3 Transit Site Support

Starting with Cisco IOS XE Release 3.15S and Cisco IOS Release 15.5(2)T release, Performance Routing version 3 (PfRv3) supports multiple data centers at the hub site. The multi-data center or the transit site support feature enables service providers to scale their network infrastructure, and load-balance the traffic when required.

Feature Information for PfRv3 Transit Site Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 333: Feature Information for PfRv3 Transit Site Support

Feature Name	Releases	Feature Information
PfRv3 Transit Site Support	15.5(2)T Cisco IOS XE Release 3.15S	The PfRv3 Transit Site Support feature enables service providers to configure multiple-data centers at the hub site. The following commands were modified by this feature: **master** (domain VRF configuration), **domain** (interface configuration).

Prerequisites for PfRv3 Transit Site Support

- Upgrade all branch sites, hub, and transit sites with latest Cisco IOS image to enable transit site support feature.

Restrictions for PfRv3 Transit Site Support

- Multiple next hops are supported only on hub or transit hub.
- Basic tunnel function is not supported between an old Cisco IOS release version and a new version, if transit site support is enabled.
- Hub sites must be connected by a Layer 3 routed link, which provides primary routing between the hub sites. Routing between hub sites over the DMVPN network is not supported

Information About PfRv3 Transit Site Support

Information About Transit Site Support

The multi-data center or the transit site support feature enables service providers to scale their network infrastructure, and load-balance the traffic when required. The multi-data center support enables all the hub sites to be connected with all the branch sites in an enterprise network. For example, in a use case scenario, an organization with two data centers and a single branch site, the branch site can communicate with the master-hub controller through the two next-hops (hub-branch routers) located at the hub site. If one hub-border router is down, then the branch site can still communicate through the second hub-border router. To differentiate the traffic from different hub-border routers, a path-id is configured on each interface of every channel. The branch router determines the inbound traffic based on the path-id of hub-branch routers. A path-id is a unique 32-bit number for a path between two sites.

PfRv3 Transit Site Use Case Scenarios

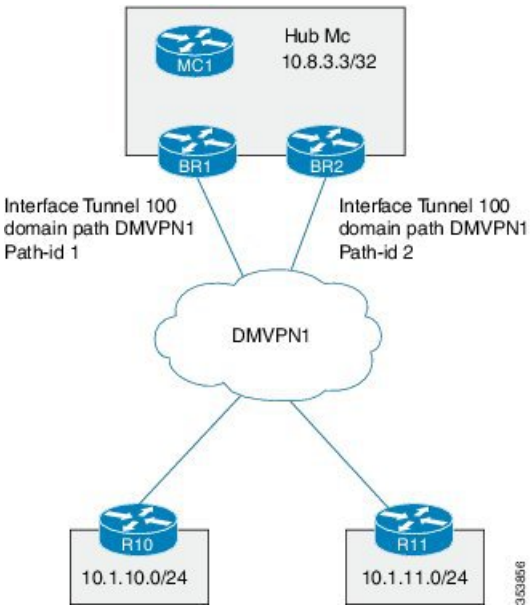
The transit site support feature supports the following use case scenarios:

- Single data center with multiple borders
- Dual data center with multiple borders
- Dual data center with same prefix

Single Data Center with Multiple Borders

In the following illustration, spoke A (R10) is connected to two (BR1 and BR2) DMVPN hubs in a single Dynamic Multipoint VPN (DMVPN) domain. There are two paths and two next-hops to the hub site from the spoke A. To differentiate traffic from different ISP paths, a path-id is added on each domain path. Use the **domain** *domain-name* **path** *path-name* **path-id** command to configure the path-ids.

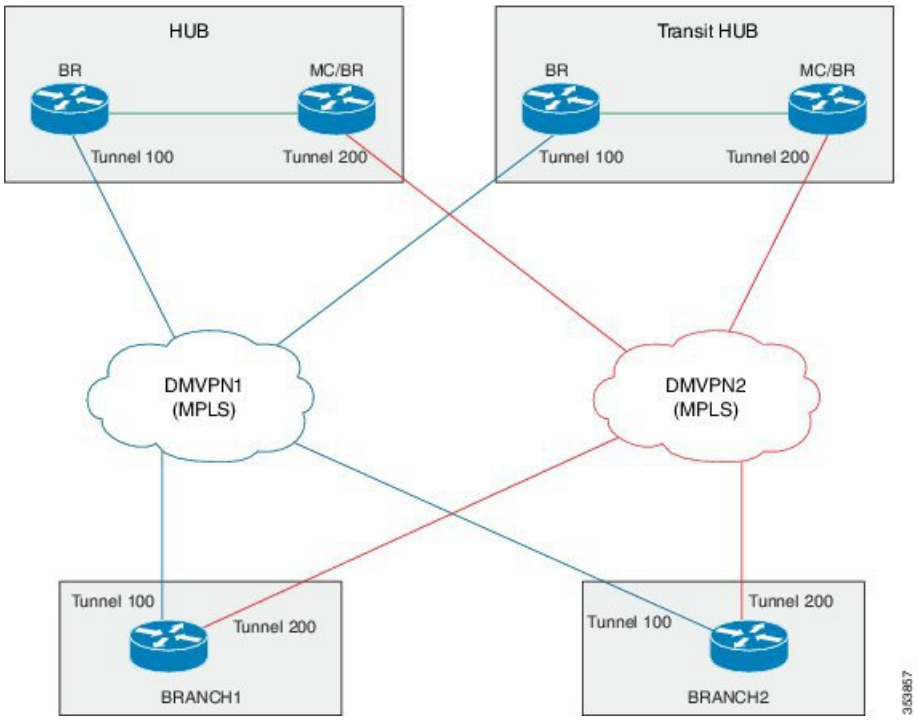
Figure 228: Single Data Center with Multiple Borders



Dual Data Center with Multiple Borders

In the following illustration, the two data centers are connected to all the branch sites. You can use both the data centers in active mode and use separate prefixes for both the data centers. To differentiate the traffic originating from different data centers, a transit-id is assigned to each data center. The valid range for a transit-id is from 1 to 62. By default, 0 is assigned to the master hub. Use the **master transit** command to configure the transit-id.

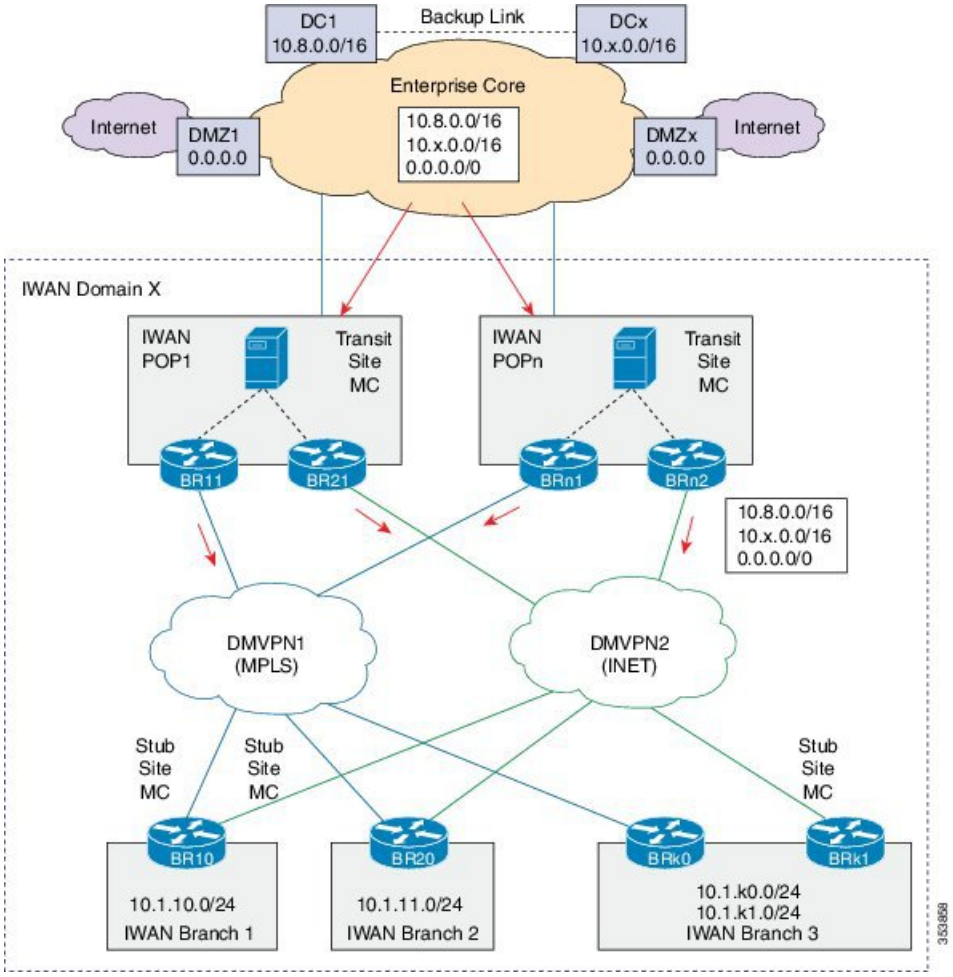
Figure 229: Dual Data Center with Multiple Borders



Dual Data Center with Same Prefix

In the following illustration, two data centers are connected to all the branch sites. However, in this scenario both the data centers are active and load-balance the traffic. If one data center is down, then traffic is routed through the other data center. Both the data centers share the same prefix.

Figure 230: Dual Data Center with Same Prefix



How to Configure Transit Site Support

Configuring Transit Hub

Before you begin

Configure the primary hub before configuring the transit hub.

Note In the current release, transit hub support is available only on Cisco ASR 1000 Series Aggregation Services Routers and Cisco 4000 Series Integrated Services Routers.

Note All policies are configured on the primary hub-master controller.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface-number*
4. **exit**
5. **domain** {*domain-name* | **default**}
6. **vrf** {*vrf-name* | **default**}
7. **master transit** *pop-id*
8. **source-interface loopback** *interface-number*
9. **site-prefixes prefix-list** *site -list*
10. **hub** *ip-address*
11. **exit**
12. **end**
13. (Optional) **show domain** *domain-name* **master status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface-number* **Example:** `Device(config)# interface Loopback0`	Enters interface configuration mode.
Step 4	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 5	**domain** {*domain-name*	**default**} **Example:**	Enters domain configuration mode.

	Command or Action	Purpose
	`Device(config)# domain default`	**Note** You can either configure a default domain or define a specific domain for the transit hub configuration. If you are defining a specific domain, for example "domain-cisco", you must configure the same domain for all devices for PfRv3 configuration.
Step 6	**vrf** {*vrf-name* | **default**} **Example:** `Device(config-domain)# vrf default`	Configures default Virtual Routing and Forwarding (VRF) instances for the default or specific domain.
Step 7	**master transit** *pop-id* **Example:** `Device(config-domain-vrf)# master transit 1`	Enters master-controller configuration mode and configures the master as a transit hub. The valid range for a pop-id is from 1 to 62.
Step 8	**source-interface loopback** *interface-number* **Example:** `Device(config-domain-vrf-mc)# source-interface Loopback0`	Configures the loopback used as a source for peering with other sites or master controller. **Note** The source-interface loopback also serves as a site ID of a particular site (hub or branch) on the master controller.
Step 9	**site-prefixes prefix-list** *site -list* **Example:** `Device(config-domain-vrf-mc)# site-prefixes prefix-list Data_Center_1`	Configures the prefix-list containing list of site prefixes. **Note** You must configure the static-site prefix list for a hub and transit sites.
Step 10	**hub** *ip-address* **Example:** `Device(config-domain-vrf-mc)# hub 10.8.3.3`	Configures the hub for the transit site.
Step 11	**exit** **Example:** `Device(config-domain-vrf-mc)# exit`	Exits from master controller configuration mode and returns to domain configuration mode. **Note** Exit from domain configuration mode and enter in global configuration mode using the **exit** command.
Step 12	**end** **Example:** `Device(config)# end`	Exits configuration mode and returns to privileged EXEC mode.
Step 13	(Optional) **show domain** *domain-name* **master status** **Example:** `Device# show domain one master status`	Use this show command to display the status of a master controller.

Configuring Transit Site Border Routers

Note In Cisco IOS XE Release 3.15S and Cisco IOS Release 15.5(2)T release, the transit site support is available only on Cisco ASR 1000 Series Aggregation Services Routers and Cisco 4000 Series Integrated Services Routers.

In a transit site support scenario, you must configure hub-border routers with the following:

- The source interface of the border router
- The IP address of the hub-master controller
- The domain path name on external interfaces
- The domain path ID for each external interface

To configure multiple hub-border routers to the same ISP path, perform the following task on each hub-border router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *interface-number*
4. **ip address** *ip-address-mask*
5. **exit**
6. **domain** {*domain-name* | **default**}
7. **vrf** {*vrf-name* | **default**}
8. **border**
9. **source-interface loopback** *interface-number*
10. **master** *ip-address*
11. **exit**
12. **exit**
13. **exit**
14. **interface** *tunnel-name*
15. **ip address** *ip-address mask*
16. **description** *description-line*
17. **domain** *domain-name* **path** *path-name* **path-id** *path-id*
18. **end**
19. (Optional) **show domain** *domain-name* **border status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *interface-number* **Example:** `Device(config)# interface Loopback0`	Enters interface configuration mode.
Step 4	**ip address** *ip-address-mask* **Example:** `Device(config-if)# ip address 10.9.4.4 255.255.255.255`	Configures an IP address for an interface on the hub-border router (Border Router 1).
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**domain** {*domain-name* | **default**} **Example:** `Device(config)# domain default`	Enters domain configuration mode.
Step 7	**vrf** {*vrf-name* | **default**} **Example:** `Device(config-domain)# vrf default`	Configures Virtual Routing and Forwarding (VRF) for the default domain. **Note** You can configure specific VRF definition for the hub-border configuration.
Step 8	**border** **Example:** `Device(config-domain-vrf)# border`	Enters border configuration mode and configures the device as border.
Step 9	**source-interface loopback** *interface-number* **Example:** `Device(config-domain-vrf-br)# source-interface Loopback0`	Configures the loopback used as a source for peering with other sites or master controller.
Step 10	**master** *ip-address* **Example:** `Device(config-domain-vrf-br)# master 10.9.3.3`	Configures the IP address of the hub-master controller.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Device(config-domain-vrf-br)# exit`	Exits border configuration mode and enters VRF configuration mode.
Step 12	**exit** **Example:** `Device(config-domain-vrf)# exit`	Exits VRF configuration mode and enters domain configuration mode.
Step 13	**exit** **Example:** `Device(config-domain)# exit`	Exits domain configuration mode and enters global configuration mode.
Step 14	**interface** *tunnel-name* **Example:** `Device(config)# interface Tunnel100`	Enters interface configuration mode.
Step 15	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.100.84 255.255.255.0`	Configures an IP address for the tunnel interface.

Step 16	**description** *description-line*
	Example:
	`Device1(config-if)# description primary path`
	`Device2(config-if)# description secondary path`
		Configures a description to associate with an ISP path.

Step 17	**domain** *domain-name* **path** *path-name* **path-id** *path-id* **Example:** `Device(config-if)# domain default path MPLS path-id 1`	Configures the Internet Service Provider (ISP) associated with the domain and the path. There are two types of external interfaces, enterprise link such as DMVPN tunnel interface and internet -bound interface. Multiple next hop is supported only on DMVPN tunnel interfaces. The path-id is a unique identifier for each path in a domain. Valid values for a path-id are from 1 to 62. We recommend using front VRF on the tunnel interface for enterprise links. **Note** You can configure multiple ISPs. If you are defining specific domain name for example, domain_cisco, you must specify the same domain name for configuring ISP paths. You must assign a unique path-id for all the paths that are connected from hub-border routers to the same ISP domain.
Step 18	**end** **Example:**	Exits interface configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
Step 19	(Optional) **show domain** *domain-name* **border status** **Example:** `Device# show domain default border status`	Use this show command to display the status of a border router.

What to do next

Verifying PfRv3 Transit Site Support

Verifying PfRv3 Transit Site Support

The **show** commands can be entered in any order.

Before you begin

Configure multiple DMVPN paths from hub-border routers or from transit-hub border routers.

SUMMARY STEPS

1. **show domain** *domain-name* **master channels**
2. **show domain** *domain-name* **border channel**
3. **show domain** *domain-name* **master site-prefix**
4. **show domain** *domain-name* **border site-prefix**
5. **show domain** *domain-name* **master channels dst-site-id** *destination-site-id*

DETAILED STEPS

Step 1 **show domain** *domain-name* **master channels**

Displays channel information of the hub-master controller.

Example:

```
HubMC# show domain default master channels

------------------------------------------------------------------------------------------------
Channel Id: 8  Dst Site-Id: 10.2.11.11  Link Name: MPLS  DSCP: default [0] pfr-label: 0:0 | 2:30
[0x21E] TCs: 0
  Channel Created: 03:19:14 ago
  Provisional State: Initiated and open
  Operational state: Available but unreachable
  Channel to hub: FALSE
  Interface Id: 11
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 0 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  ODE Stats Bucket Number: 1
   Last Updated  : 00:00:21 ago
    Packet Count  : 0
    Byte Count    : 0
```

```
    One Way Delay  : N/A
    Loss Rate Pkts : N/A
    Loss Rate Bytes: N/A
    Jitter Mean    : N/A
    Unreachable   : TRUE
  ODE Stats Bucket Number: 2
   Last Updated  : 00:00:52 ago
    Packet Count  : 0
    Byte Count    : 0
    One Way Delay  : N/A
    Loss Rate Pkts : N/A
    Loss Rate Bytes: N/A
    Jitter Mean    : N/A
    Unreachable   : TRUE
   TCA Statistics:
      Received:355 ; Processed:354 ; Unreach_rcvd:355
  Latest TCA Bucket
   Last Updated  : 00:00:21 ago
    Local unreachable TCA received(Check for stale TCA 00:00:09 later)
.
.
.
------------------------------------------------------------------------------------------
```

Step 2 **show domain** *domain-name* **border channel**

Displays the information of border router channels at the hub site.

Example:

```
HubBR# show domain default border channels

------------------------------------------------------------------------------------------
Border Smart Probe Stats:

 Smart probe parameters:
   Source address used in the Probe: 10.2.10.10
   Unreach time: 1000 ms
   Probe source port: 18000
   Probe destination port: 19000
   Interface Discovery: ON
   Probe freq for channels with traffic :10 secs
   Discovery Probes: OFF
   Number of transit probes consumed :29
   Number of transit probes re-routed: 0
   DSCP's using this: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17]
 [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37]
 [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57]
 [58] [59] [60] [61] [62] [63] [64]
   All the other DSCPs use the default interval: 10 secs

Channel id: 20
  Channel create time: 06:42:54 ago
  Site id : 10.2.11.11
  DSCP : default[0]
  Service provider : MPLS
  Pfr-Label : 0:0 | 0:0 [0x0]
  exit path-id: 0
  Exit path-id sent on wire: 0
  Number of Probes sent : 77407
  Number of Probes received : 75949
  Last Probe sent : 00:00:00 ago
  Last Probe received : 00:00:00 ago
  Channel state : Initiated and open
  Channel next_hop : 10.0.100.11
```

```
  RX Reachability : Reachable
  TX Reachability : Reachable
  Channel is sampling 0 flows
  Channel remote end point: 10.0.100.11
  Channel to hub: FALSE
  Version: 3
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Probe freq with traffic : 1 in 10000 ms

.
.
.
------------------------------------------------------------------------------------------------
```

Step 3 **show domain** *domain-name* **master site-prefix**

Displays the details of site-prefixes configured to the master hub.

Example:

```
HubMC# show domain default master site-prefix

------------------------------------------------------------------------------------------------
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 11:28:29.421 CET Tue Mar 17 2015

  Change will be published between 5-60 seconds
  Next Publish 00:33:03 later
  Prefix DB Origin: 10.9.3.3
  Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id              Site-prefix          Last Updated         DC Bitmap  Flag
------------------------------------------------------------------------------
10.2.10.10            10.1.10.0/24         01:25:15 ago         0x0         S
10.2.11.11            10.1.11.0/24         01:25:19 ago         0x0         S
10.2.10.10            10.2.10.10/32        01:25:15 ago         0x0         S
10.2.11.11            10.2.11.11/32        01:25:19 ago         0x0         S
10.2.12.12            10.2.12.12/32        01:28:54 ago         0x0         S
10.8.3.3              10.8.3.3/32          01:28:47 ago         0x1         S
10.9.3.3              10.8.0.0/16          01:28:47 ago         0x5         C,M
10.8.3.3              10.8.0.0/16          01:28:47 ago         0x5         C,M
10.9.3.3              10.9.3.3/32          03:29:04 ago         0x4         L
10.9.3.3              10.9.0.0/16          01:28:47 ago         0x5         C,M
10.8.3.3              10.9.0.0/16          01:28:47 ago         0x5         C,M
255.255.255.255      *10.0.0.0/8           01:28:47 ago         0x1         S,T

------------------------------------------------------------------------------------------------
```

Step 4 **show domain** *domain-name* **border site-prefix**

Displays the details of site-prefixes configured on the border.

Example:

```
HubBR# show domain default border site-prefix

------------------------------------------------------------------------------------------------

Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id              Site-prefix          Last Updated         DC Bitmap  Flag
------------------------------------------------------------------------------
10.2.10.10            10.1.10.0/24         00:36:58 ago         0x0         S
10.2.11.11            10.1.11.0/24         00:37:02 ago         0x0         S
```

```
10.2.10.10            10.2.10.10/32        00:36:58 ago         0x0         S
10.2.11.11            10.2.11.11/32        00:37:02 ago         0x0         S
10.2.12.12            10.2.12.12/32        00:40:37 ago         0x0         S
10.8.3.3              10.8.3.3/32          00:40:29 ago         0x1         S
10.9.3.3              10.8.0.0/16          00:38:40 ago         0x5         S,C,M
10.8.3.3              10.8.0.0/16          00:38:40 ago         0x5         S,C,M
10.9.3.3              10.9.3.3/32          00:38:40 ago         0x4         S
10.9.3.3              10.9.0.0/16          00:38:40 ago         0x5         S,C,M
10.8.3.3              10.9.0.0/16          00:38:40 ago         0x5         S,C,M
255.255.255.255      *10.0.0.0/8           00:40:29 ago         0x1         S,T
-----------------------------------------------------------------------------------------------
```

Step 5 **show domain** *domain-name* **master channels dst-site-id** *destination-site-id*

Displays the details of destination site-ids configured with hub-master controller.

Note Use this command on a spoke or a branch device to view the details of the destination site-ids.

Example:

```
BR# show domain default master channels dst-site-id 10.8.3.3

-----------------------------------------------------------------------------------------------

Legend: * (Value obtained from Network delay:)

Channel Id: 27  Dst Site-Id: 10.8.3.3  Link Name: INET  DSCP: default [0] pfr-label: 0:20 | 0:0
[0x140000] TCs: 0
  Channel Created: 01:16:34 ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Interface Id: 12
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 5 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    10.8.3.3/32 (Active)
    10.8.0.0/16 (Active)
    10.9.0.0/16 (Standby)
  ODE Stats Bucket Number: 1
   Last Updated  : 00:00:24 ago
    Packet Count  : 562
Byte Count    : 47208
    One Way Delay : 71 msec*
    Loss Rate Pkts: 0.0 %
    Loss Rate Byte: 0.0 %
    Jitter Mean   : 619 usec
    Unreachable   : FALSE
  ODE Stats Bucket Number: 2
   Last Updated  : 00:00:54 ago
    Packet Count  : 558
    Byte Count    : 46872
    One Way Delay : 55 msec*
    Loss Rate Pkts: 0.0 %
    Loss Rate Byte: 0.0 %
    Jitter Mean   : 556 usec
    Unreachable   : FALSE
   TCA Statistics:
      Received:133 ; Processed:133 ; Unreach_rcvd:0
  Latest TCA Bucket
   Last Updated  : 00:00:24 ago
```

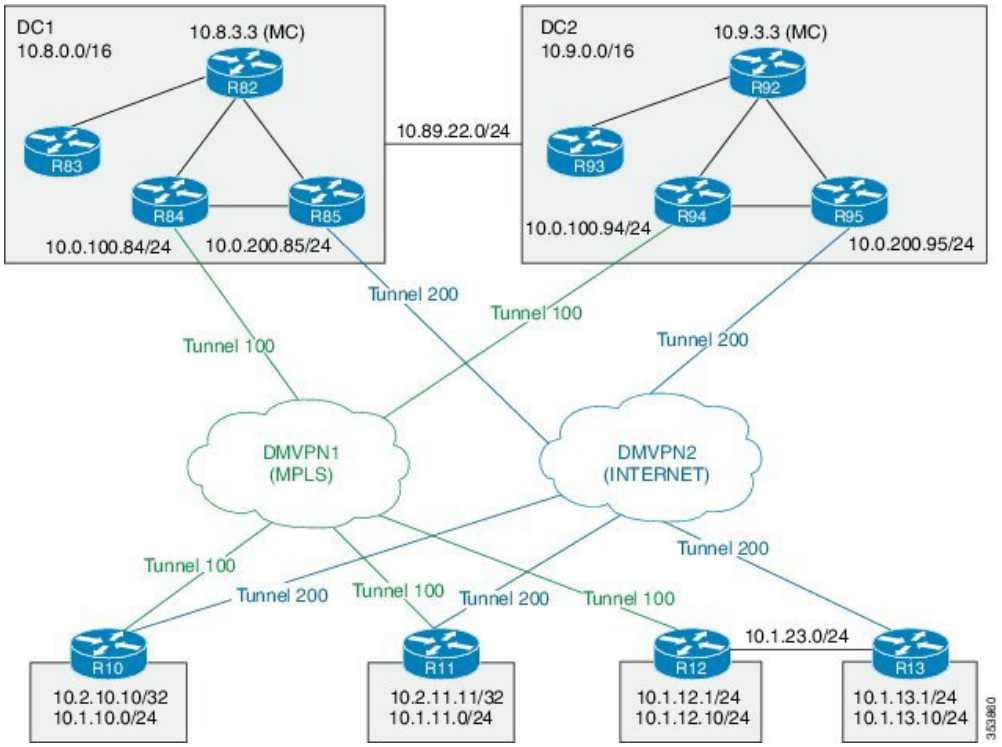
```
    One Way Delay : 71 msec*
    Loss Rate Pkts: NA
    Loss Rate Byte: NA
    Jitter Mean   : NA
    Unreachability: FALSE
.
.
.
------------------------------------------------------------------------------------------
```

Configuration Examples for PfRv3 Transit Site Support

Example: Configuring Transit Site Support

In this use case scenario, an enterprise organization has two data centers with multiple-border routers connected to the same ISP domain. The branch-border routers can reach the hub-master controller through multiple next-hops.

Figure 231: PfRv3 Transit Hub Topology



In this example, the following routers are used:

- Hub Master Controller — Cisco ASR 1002-X router configured with an embedded services processor (ESP) default bandwidth of 5 Gbps upgradable with software licensing options to 10 Gbps, 20 Gbps, and 36 Gbps.

- Hub Border Routers — Cisco ASR 1000 Series Embedded Services Processor 2
- Branch Routers — Cisco 4451X Integrated Services Router.

Example: Configuring Data Center 1 (DC1) Devices

Configure the interfaces on master hub controller (R82)

```
HubMC> enable
HubMC# configure terminal
HubMC(config)# interface Loopback0
HubMC(config-if)# ip address 10.8.3.3 255.255.255.255
HubMC(config-if)# exit
```

Configure the device as hub-master controller

```
HubMC(config)# domain default
HubMC(config-domain)# vrf default
HubMC(config-domain-vrf)# master hub
HubMC(config-domain-vrf-mc)# source-interface Loopback0
HubMC(config-domain-vrf-mc)# enterprise-prefix prefix-list ENTERPRISE_PREFIX
HubMC(config-domain-vrf-mc)# site-prefixes prefix-list DC1_PREFIX
HubMC(config-domain-vrf-mc)# exit
```

Configure IP prefix-lists

```
HubMC(config)# ip prefix-list DC1_PREFIX seq 10 permit 10.8.0.0/16
HubMC(config)# ip prefix-list DC1_PREFIX seq 10 permit 10.9.0.0/16
HubMC(config)# ip prefix-list ENTERPRISE_PREFIX seq 10 permit 10.0.0.0/8
```

Configure domain policies on hub master controller

```
HubMC(config)# domain default
HubMC(config-domain)# vrf default
HubMC(config-domain-vrf)# master hub
HubMC(config-domain-vrf-mc)# source-interface Loopback0
HubMC(config-domain-vrf-mc)# site-prefixes prefix-list DC1_PREFIX
HubMC(config-domain-vrf-mc)# load-balance
HubMC(config-domain-vrf-mc)# enterprise-prefix  prefix-list ENTERPRISE_PREFIX

HubMC(config-domain-vrf-mc)# class VOICE sequence 10
HubMC(config-domain-vrf-mc-class)# match dscp ef policy custom
HubMC(config-domain-vrf-mc-class-type)# priority 2 loss threshold 5
HubMC(config-domain-vrf-mc-class-type)# priority 1 one-way-delay  threshold 150
HubMC(config-domain-vrf-mc-class-type)# exit
HubMC(config-domain-vrf-mc-class)# path-preference MPLS fallback INET
HubMC(config-domain-vrf-mc-class)# exit

HubMC(config-domain-vrf-mc)# class VIDEO sequence 20
HubMC(config-domain-vrf-mc-class)# match dscp af41 policy custom
HubMC(config-domain-vrf-mc-class-type)# priority 2 loss threshold 5
HubMC(config-domain-vrf-mc-class-type)# priority 1 one-way-delay  threshold 150
HubMC(config-domain-vrf-mc-class-type)# exit
HubMC(config-domain-vrf-mc-class)# match dscp cs4 policy custom
HubMC(config-domain-vrf-mc-class-type)# priority 2 loss threshold 5
HubMC(config-domain-vrf-mc-class-type)# priority 1 one-way-delay  threshold 150
HubMC(config-domain-vrf-mc-class-type)# exit
HubMC(config-domain-vrf-mc-class)# path-preference INET fallback MPLS
HubMC(config-domain-vrf-mc-class)# exit


HubMC(config-domain-vrf-mc)# class CRITICAL sequence 30
```

```
HubMC(config-domain-vrf-mc-class)# match dscp af31 policy custom
HubMC(config-domain-vrf-mc-class-type)# priority 2 loss threshold 10
HubMC(config-domain-vrf-mc-class-type)# priority 1 one-way-delay threshold 600
HubMC(config-domain-vrf-mc-class-type)# exit
HubMC(config-domain-vrf-mc-class)# path-preference MPLS fallback INET
HubMC(config-domain-vrf-mc)# class DEFAULT sequence 100
HubMC(config-domain-vrf-mc-class)# match dscp default policy custom
HubMC(config-domain-vrf-mc-class-type)# priority 2 loss threshold 5
HubMC(config-domain-vrf-mc-class-type)# priority 1 one-way-delay threshold 50
HubMC(config-domain-vrf-mc-class-type)# priority 3 jitter threshold 200000
HubMC(config-domain-vrf-mc-class-type)# exit
```

Configure hub border routers on DC1 (R84)

```
BR84> enable
BR84# configure terminal
BR84(config)# interface Loopback0
BR84(config-if)# ip address 10.8.4.4 255.255.255.255
BR84(config-if)exit
```

Configure the device as border router (BR84)

```
BR84(config)# domain default
BR84(config-domain)# vrf default
BR84(config-domain-vrf)# border
BR84(config-domain-vrf-br)# source-interface Loopback0
BR84(config-domain-vrf-br)# master 10.8.3.3
BR84(config-domain-vrf-br)# exit
```

Configure tunnel from BR84 to DMVPN1 (MPLS)Link

```
BR84(config)# interface Tunnel100
BR84(config-if)# bandwidth 100000
BR84(config-if)# ip address 10.0.100.84 255.255.255.0
BR84(config-if)# no ip redirects
BR84(config-if)# ip mtu 1400
BR84(config-if)# ip nhrp authentication cisco
BR84(config-if)# ip nhrp map multicast dynamic
BR84(config-if)# ip nhrp network-id 1
BR84(config-if)# ip nhrp holdtime 60
BR84(config-if)# ip nhrp redirect
BR84(config-if)# ip tcp adjust-mss 1360
BR84(config-if)# load-interval 30
BR84(config-if)# delay 1000
BR84(config-if)# tunnel source Ethernet0/1
BR84(config-if)# tunnel mode gre multipoint
BR84(config-if)# tunnel key 100
BR84(config-if)# tunnel vrf IWAN-TRANSPORT-1
BR84(config-if)# domain path MPLS path-id 10
```

Configure hub border routers on DC1 (R85)

```
BR85> enable
BR85# configure terminal
BR85(config)# interface Loopback0
BR85(config-if)# ip address 10.8.5.5 255.255.255.255
BR85(config-if)exit
```

Configure the device as border router (BR85)

```
BR85(config)# domain default
BR85(config-domain)# vrf default
BR85(config-domain-vrf)# border
BR85(config-domain-vrf-br)# source-interface Loopback0
```

```
BR85(config-domain-vrf-br)# master 10.8.3.3
BR85(config-domain-vrf-br)# exit
```

Configure tunnel from BR84 to DMVPN2 (INET)Link

```
BR85(config)# interface Tunnel200
BR85(config-if)# bandwidth 5000
BR85(config-if)# ip address 10.0.200.85 255.255.255.0
BR85(config-if)# no ip redirects
BR85(config-if)# ip mtu 1400
BR85(config-if)# ip nhrp authentication cisco
BR85(config-if)# ip nhrp map multicast dynamic
BR85(config-if)# ip nhrp network-id 2
BR85(config-if)# ip nhrp holdtime 60
BR85(config-if)# ip nhrp redirect
BR85(config-if)# ip tcp adjust-mss 1360
BR85(config-if)# load-interval 30
BR85(config-if)# delay 1000
BR85(config-if)# tunnel source Ethernet0/1
BR85(config-if)# tunnel mode gre multipoint
BR85(config-if)# tunnel key 200
BR85(config-if)# tunnel vrf IWAN-TRANSPORT-2
BR85(config-if)# domain path INET path-id 20
```

Example: Configuring Data Center 2 (DC2) Devices

Configure the interfaces on master hub controller (R92)

```
HubMC> enable
HubMC# configure terminal
HubMC(config)# interface Loopback0
HubMC(config-if)# ip address 10.9.3.3 255.255.255.255
HubMC(config-if)# exit
```

Configure the device as transit-hub master controller

```
HubMC(config)# domain default
HubMC(config-domain)# vrf default
HubMC(config-domain-vrf)# master transit 2
HubMC(config-domain-vrf-mc)# source-interface Loopback0
HubMC(config-domain-vrf-mc)# site-prefixes prefix-list DC2_PREFIX
HubMC(config-domain-vrf-mc)# hub 10.8.3.3
HubMC(config-domain-vrf-mc)# exit
```

Configure IP prefix-lists

```
HubMC(config)# ip prefix-list DC2_PREFIX seq 10 permit 10.9.0.0/16
HubMC(config)# ip prefix-list DC2_PREFIX seq 20 permit 10.8.0.0/16
HubMC(config)# ip prefix-list ENTERPRISE_PREFIX seq 10 permit 10.0.0.0/8
```

Configure hub border routers on DC2 (R94)

```
BR94> enable
BR94# configure terminal
BR94(config)# interface Loopback0
BR94(config-if)# ip address 10.9.4.4 255.255.255.255
BR94(config-if)exit
```

Configure the device as border router (BR94)

```
BR94(config)# domain default
BR94(config-domain)# vrf default
BR94(config-domain-vrf)# border
BR94(config-domain-vrf-br)# source-interface Loopback0
```

```
BR94(config-domain-vrf-br)# master 10.9.3.3
BR94(config-domain-vrf-br)# exit
```

Configure tunnel from BR94 to DMVPN1 (MPLS)Link

```
BR94(config)# interface Tunnel100
BR94(config-if)# bandwidth 1000
BR94(config-if)# ip address 10.0.100.94 255.255.255.0
BR94(config-if)# no ip redirects
BR94(config-if)# ip mtu 1400
BR94(config-if)# ip nhrp authentication cisco
BR94(config-if)# ip nhrp map multicast dynamic
BR94(config-if)# ip nhrp network-id 1
BR94(config-if)# ip nhrp holdtime 60
BR94(config-if)# ip nhrp redirect
BR94(config-if)# ip tcp adjust-mss 1360
BR94(config-if)# load-interval 30
BR94(config-if)# delay 1000
BR94(config-if)# tunnel source Ethernet0/1
BR94(config-if)# tunnel mode gre multipoint
BR94(config-if)# tunnel key 100
BR94(config-if)# tunnel vrf IWAN-TRANSPORT-1
BR94(config-if)# domain path MPLS path-id 30
```

Configure hub border routers on DC2 (R95)

```
BR95> enable
BR95# configure terminal
BR95(config)# interface Loopback0
BR95(config-if)# ip address 10.9.5.5 255.255.255.255
BR95(config-if)exit
```

Configure the device as border router (BR95)

```
BR95(config)# domain default
BR95(config-domain)# vrf default
BR95(config-domain-vrf)# border
BR95(config-domain-vrf-br)# source-interface Loopback0
BR95(config-domain-vrf-br)# master 10.9.3.3
BR95(config-domain-vrf-br)# exit
```

Configure tunnel from BR95 to DMVPN2 (INET)Link

```
BR95(config)# interface Tunnel200
BR95(config-if)# bandwidth 1000
BR95(config-if)# ip address 10.0.200.95 255.255.255.0
BR95(config-if)# no ip redirects
BR95(config-if)# ip mtu 1400
BR95(config-if)# ip nhrp authentication cisco
BR95(config-if)# ip nhrp map multicast dynamic
BR95(config-if)# ip nhrp network-id 2
BR95(config-if)# ip nhrp holdtime 60
BR95(config-if)# ip nhrp redirect
BR95(config-if)# ip tcp adjust-mss 1360
BR95(config-if)# load-interval 30
BR95(config-if)# delay 1000
BR95(config-if)# tunnel source Ethernet0/1
BR95(config-if)# tunnel mode gre multipoint
BR95(config-if)# tunnel key 200
BR95(config-if)# tunnel vrf IWAN-TRANSPORT-2
BR95(config-if)# domain path INET path-id 40
```

Example: Configuring Branch Routers

Configure the interfaces (R10)

```
R10> enable
R10# configure terminal
R10(config)# interface Loopback0
R10(config-if)# ip address 10.2.10.10 255.255.255.255
R10(config-if)# exit
```

Configure the device as branch-master controller (R10)

```
R10(config)# domain default
R10(config-domain)# vrf default
R10(config-domain-vrf)# border
R10(config-domain-vrf-br)# source-interface Loopback0
R10(config-domain-vrf-br)# master local
R10(config-domain-vrf-br)# exit
R10(config-domain-vrf)# master branch
R10(config-domain-vrf-mc)# source-interface Loopback0
R10(config-domain-vrf-mc)# hub 10.8.3.3
```

Configure the tunnel interface and tunnel path from R10

```
R10(config)# interface Tunnel100
R10(config-if)# bandwidth 400
R10(config-if)# ip address 10.0.100.10 255.255.255.0
R10(config-if)# no ip redirects
R10(config-if)# ip mtu 1400
R10(config-if)# ip nhrp authentication cisco
R10(config-if)# ip nhrp map multicast dynamic
R10(config-if)# ip nhrp network-id 1
R10(config-if)# ip nhrp holdtime 60
R10(config-if)# ip nhrp nhs 10.0.100.84 nbma 172.16.84.4 multicast
R10(config-if)# ip nhrp nhs 10.0.100.94 nbma 172.16.94.4 multicast
R10(config-if)# ip nhrp registration no-unique
R10(config-if)# ip nhrp registration timeout 60
R10(config-if)# ip nhrp shortcut
R10(config-if)# ip nhrp redirect
R10(config-if)# ip tcp adjust-mss 1360
R10(config-if)# load-interval 30
R10(config-if)# delay 1000
R10(config-if)# no nhrp route-watch
R10(config-if)# if-state nhrp
R10(config-if)# tunnel source Ethernet0/1
R10(config-if)# tunnel mode gre multipoint
R10(config-if)# tunnel key 100
R10(config-if)# tunnel vrf IWAN-TRANSPORT-1

R10(config)# interface Tunnel200
R10(config-if)# bandwidth 5000
R10(config-if)# ip address 10.0.200.10 255.255.255.0
R10(config-if)# no ip redirects
R10(config-if)# ip mtu 1400
R10(config-if)# ip nhrp authentication cisco
R10(config-if)# ip nhrp map multicast dynamic
R10(config-if)# ip nhrp network-id 2
R10(config-if)# ip nhrp holdtime 600
R10(config-if)# ip nhrp nhs 10.0.200.85 nbma 172.16.85.5 multicast
R10(config-if)# ip nhrp nhs 10.0.200.95 nbma 172.16.95.5 multicast
R10(config-if)# ip nhrp registration no-unique
R10(config-if)# ip nhrp registration timeout 60
```

```
R10(config-if)# ip nhrp shortcut
R10(config-if)# ip nhrp redirect
R10(config-if)# ip tcp adjust-mss 1360
R10(config-if)# load-interval 30
R10(config-if)# delay 1000
R10(config-if)# no nhrp route-watch
R10(config-if)# if-state nhrp
R10(config-if)# tunnel source Ethernet0/2
R10(config-if)# tunnel mode gre multipoint
R10(config-if)# tunnel key 200
R10(config-if)# tunnel vrf IWAN-TRANSPORT-2
```

Configure the interfaces (R11)

```
R11> enable
R11# configure terminal
R11(config)# interface Loopback0
R11(config-if)# ip address 10.2.11.11 255.255.255.255
R11(config-if)# exit
```

Configure the device as branch master controller (R11)

```
R11(config)# domain default
R11(config-domain)# vrf default
R11(config-domain-vrf)# border
R11(config-domain-vrf-br)# source-interface Loopback0
R11(config-domain-vrf-br)# master local
R11(config-domain-vrf-br)# exit
R11(config-domain-vrf)# master branch
R11(config-domain-vrf-mc)# source-interface Loopback0
R11(config-domain-vrf-mc)# hub 10.8.3.3
```

Configure the tunnel interface and tunnel path from R11

```
R11(config)# interface Tunnel100
R11(config-if)# bandwidth 2000
R11(config-if)# ip address 10.0.100.11 255.255.255.0
R11(config-if)# no ip redirects
R11(config-if)# ip mtu 1400
R11(config-if)# ip nhrp authentication cisco
R11(config-if)# ip nhrp map multicast dynamic
R11(config-if)# ip nhrp network-id 1
R11(config-if)# ip nhrp holdtime 60
R11(config-if)# ip nhrp nhs 10.0.100.84 nbma 172.16.84.4 multicast
R11(config-if)# ip nhrp nhs 10.0.100.94 nbma 172.16.94.4 multicast
R11(config-if)# ip nhrp registration no-unique
R11(config-if)# ip nhrp registration timeout 60
R11(config-if)# ip nhrp shortcut
R11(config-if)# ip nhrp redirect
R11(config-if)# ip tcp adjust-mss 1360
R11(config-if)# load-interval 30
R11(config-if)# delay 1000
R11(config-if)# no nhrp route-watch
R11(config-if)# if-state nhrp
R11(config-if)# tunnel source Ethernet0/1
R11(config-if)# tunnel mode gre multipoint
R11(config-if)# tunnel key 100
R11(config-if)# tunnel vrf IWAN-TRANSPORT-1

R11(config)# interface Tunnel200
R11(config-if)# bandwidth 5000
R11(config-if)# ip address 10.0.200.11 255.255.255.0
R11(config-if)# no ip redirects
R11(config-if)# ip mtu 1400
```

```
R11(config-if)# ip nhrp authentication cisco
R11(config-if)# ip nhrp map multicast dynamic
R11(config-if)# ip nhrp network-id 2
R11(config-if)# ip nhrp holdtime 600
R11(config-if)# ip nhrp nhs 10.0.200.85 nbma 172.16.85.5 multicast
R11(config-if)# ip nhrp nhs 10.0.200.95 nbma 172.16.95.5 multicast
R11(config-if)# ip nhrp registration no-unique
R11(config-if)# ip nhrp registration timeout 60
R11(config-if)# ip nhrp shortcut
R11(config-if)# ip nhrp redirect
R11(config-if)# ip tcp adjust-mss 1360
R11(config-if)# load-interval 30
R11(config-if)# delay 1000
R11(config-if)# no nhrp route-watch
R11(config-if)# if-state nhrp
R11(config-if)# tunnel source Ethernet0/2
R11(config-if)# tunnel mode gre multipoint
R11(config-if)# tunnel key 200
R11(config-if)# tunnel vrf IWAN-TRANSPORT-2
```

Configure the interfaces (R12)

```
R12> enable
R12# configure terminal
R12(config)# interface Loopback0
R12(config-if)# ip address 10.2.12.12 255.255.255.255
R12(config-if)# exit
```

Configure the device as branch-master controller (R12)

```
R12(config)# domain default
R12(config-domain)# vrf default
R12(config-domain-vrf)# border
R12(config-domain-vrf-br)# source-interface Loopback0
R12(config-domain-vrf-br)# master local
R12(config-domain-vrf-br)# exit
R12(config-domain-vrf)# master branch
R12(config-domain-vrf-mc)# source-interface Loopback0
R12(config-domain-vrf-mc)# hub 10.8.3.3
```

Configure the tunnel interface and tunnel path from R12

```
R12(config)# interface Tunnel100
R12(config-if)# bandwidth 400
R12(config-if)# ip address 10.0.100.12 255.255.255.0
R12(config-if)# no ip redirects
R12(config-if)# ip mtu 1400
R12(config-if)# ip nhrp authentication cisco
R12(config-if)# ip nhrp map multicast dynamic
R12(config-if)# ip nhrp network-id 1
R12(config-if)# ip nhrp holdtime 600
R12(config-if)# ip nhrp nhs 10.0.100.84 nbma 172.16.84.4 multicast
R12(config-if)# ip nhrp nhs 10.0.100.94 nbma 172.16.94.4 multicast
R12(config-if)# ip nhrp registration no-unique
R12(config-if)# ip nhrp registration timeout 60
R12(config-if)# ip nhrp shortcut
R12(config-if)# ip tcp adjust-mss 1360
R12(config-if)# load-interval 30
R12(config-if)# delay 1000
R12(config-if)# no nhrp route-watch
R12(config-if)# if-state nhrp
R12(config-if)# tunnel source Ethernet0/1
R12(config-if)# tunnel mode gre multipoint
```

```
R12(config-if)# tunnel key 100
R12(config-if)# tunnel vrf IWAN-TRANSPORT-1
```

Configure the interfaces (R13)

```
R13> enable
R13# configure terminal
R13(config)# interface Loopback0
R13(config-if)# ip address 10.2.13.13 255.255.255.255
R13(config-if)# exit
```

Configure the device as a border router with R12 as the master controller (R13)

```
R13(config)# domain default
R13(config-domain)# vrf default
R13(config-domain-vrf)# border
R13(config-domain-vrf-br)# source-interface Loopback0
R13(config-domain-vrf-br)# master 10.2.12.12
R13(config-domain-vrf-br)# exit
```

Configure the tunnel interface and tunnel path from R13

```
R13(config)# interface Tunnel200
R13(config-if)# bandwidth 400
R13(config-if)# ip address 10.0.200.13 255.255.255.0
R13(config-if)# no ip redirects
R13(config-if)# ip mtu 1400
R13(config-if)# ip nhrp authentication cisco
R13(config-if)# ip nhrp network-id 2
R13(config-if)# ip nhrp holdtime 600
R13(config-if)# ip nhrp nhs 10.0.200.85 nbma 172.16.85.5 multicast
R13(config-if)# ip nhrp nhs 10.0.100.95 nbma 172.16.95.5 multicast
R13(config-if)# ip nhrp registration no-unique
R13(config-if)# ip nhrp registration timeout 60
R13(config-if)# ip nhrp shortcut
R13(config-if)# ip tcp adjust-mss 1360
R13(config-if)# load-interval 30
R13(config-if)# delay 1000
R13(config-if)# if-state nhrp
R13(config-if)# tunnel source Ethernet0/2
R13(config-if)# tunnel mode gre multipoint
R13(config-if)# tunnel key 200
R13(config-if)# tunnel vrf IWAN-TRANSPORT-2
```

Verifying PfRv3 Transit Site Configuration

To verify the PfRv3 transit site configuration, use the following show commands in any order:

```
HubMC2# show domain default master status

---------------------------------------------------------------------------------------------
*** Domain MC Status ***

 Master VRF: Global

  Instance Type:    Transit
  POP ID:      2
  Instance id:      0
  Operational status:  Up
  Configured status:  Up
  Loopback IP Address: 10.9.3.3
  Load Balancing:
   Operational Status: Up
   Max Calculated Utilization Variance: 0%
```

```
   Last load balance attempt: 03:07:30 ago
   Last Reason:  Variance less than 20%
   Total unbalanced bandwidth:
         External links: 0 Kbps  Internet links: 0 Kbps
  Route Control: Enabled
  Mitigation mode Aggressive: Disabled
  Policy threshold variance: 20
  Minimum Mask Length: 28
  Minimum Requirement: Met

  Borders:
    IP address: 10.9.5.5
    Version: 2
    Connection status: CONNECTED (Last Updated 03:25:38 ago )
    Interfaces configured:
Name: Tunnel200 | type: external | Service Provider: INET path-id:40 | Status: UP | Zero-SLA:
 NO
          Number of default Channels: 0


    Tunnel if: Tunnel0

    IP address: 10.9.4.4
    Version: 2
    Connection status: CONNECTED (Last Updated 03:25:37 ago )
    Interfaces configured:
      Name: Tunnel100 | type: external | Service Provider: MPLS path-id:30 | Status: DOWN
    Tunnel if: Tunnel0

------------------------------------------------------------------------------------------

HubMC2# show domain default master channels

------------------------------------------------------------------------------------------
Channel Id: 8  Dst Site-Id: 10.2.11.11  Link Name: MPLS  DSCP: default [0] pfr-label: 0:0
| 2:30 [0x21E] TCs: 0
  Channel Created: 03:19:14 ago
  Provisional State: Initiated and open
  Operational state: Available but unreachable
  Channel to hub: FALSE
  Interface Id: 11
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 0 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  ODE Stats Bucket Number: 1
   Last Updated  : 00:00:21 ago
    Packet Count  : 0
    Byte Count    : 0
    One Way Delay  : N/A
    Loss Rate Pkts : N/A
    Loss Rate Bytes: N/A
    Jitter Mean    : N/A
    Unreachable   : TRUE
  ODE Stats Bucket Number: 2
   Last Updated  : 00:00:52 ago
    Packet Count  : 0
    Byte Count    : 0
    One Way Delay  : N/A
    Loss Rate Pkts : N/A
    Loss Rate Bytes: N/A
    Jitter Mean    : N/A
    Unreachable   : TRUE
```

```
   TCA Statistics:
      Received:355 ; Processed:354 ; Unreach_rcvd:355
  Latest TCA Bucket
   Last Updated  : 00:00:21 ago
    Local unreachable TCA received(Check for stale TCA 00:00:09 later)
.
.
.
-------------------------------------------------------------------------------------------------

HubMC2#  show domain default master site-capability device-capb path-id

-------------------------------------------------------------------------------------------------

Site pop id : 1
Site mc type : Transit
Border Address : 10.9.4.4
Service provider: MPLS path-id: 30 if_index: 11
Border Address : 10.9.5.5
Service provider: INET path-id: 40 if_index: 11

-------------------------------------------------------------------------------------------------


HubMC2# show domain default master site-prefix

-------------------------------------------------------------------------------------------------
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 11:28:29.421 CET Tue Mar 17 2015

  Change will be published between 5-60 seconds
  Next Publish 00:33:03 later
  Prefix DB Origin: 10.9.3.3
  Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id              Site-prefix          Last Updated         DC Bitmap  Flag
-------------------------------------------------------------------------------
10.2.10.10            10.1.10.0/24         01:25:15 ago         0x0         S
10.2.11.11            10.1.11.0/24         01:25:19 ago         0x0         S
10.2.10.10            10.2.10.10/32        01:25:15 ago         0x0         S
10.2.11.11            10.2.11.11/32        01:25:19 ago         0x0         S
10.2.12.12            10.2.12.12/32        01:28:54 ago         0x0         S
10.8.3.3              10.8.3.3/32          01:28:47 ago         0x1         S
10.9.3.3              10.8.0.0/16          01:28:47 ago         0x5         C,M
10.8.3.3              10.8.0.0/16          01:28:47 ago         0x5         C,M
10.9.3.3              10.9.3.3/32          03:29:04 ago         0x4         L
10.9.3.3              10.9.0.0/16          01:28:47 ago         0x5         C,M
10.8.3.3              10.9.0.0/16          01:28:47 ago         0x5         C,M
255.255.255.255      *10.0.0.0/8           01:28:47 ago         0x1         S,T

-------------------------------------------------------------------------------------------------

HubMC2# show domain default master policy

-------------------------------------------------------------------------------------------------

Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 11:31:10.977 CET Tue Mar 17 2015

  class VOICE sequence 10
    path-preference MPLS fallback INET
    class type: Dscp Based
      match dscp ef policy custom
        priority 2 packet-loss-rate threshold 5.0 percent
```

```
        priority 1 one-way-delay threshold 150 msec
        priority 2 byte-loss-rate threshold 5.0 percent

  class VIDEO sequence 20
    path-preference MPLS fallback INET
    class type: Dscp Based
      match dscp af41 policy custom
        priority 2 packet-loss-rate threshold 5.0 percent
        priority 1 one-way-delay threshold 150 msec
        priority 2 byte-loss-rate threshold 5.0 percent
      match dscp cs4 policy custom
        priority 2 packet-loss-rate threshold 5.0 percent
        priority 1 one-way-delay threshold 150 msec
        priority 2 byte-loss-rate threshold 5.0 percent

  class CRITICAL sequence 30
    path-preference MPLS fallback INET
    class type: Dscp Based
      match dscp af31 policy custom
        priority 2 packet-loss-rate threshold 10.0 percent
        priority 1 one-way-delay threshold 600 msec
        priority 2 byte-loss-rate threshold 10.0 percent
        Number of Traffic classes using this policy: 1

class DEFAULT0 sequence 100
    class type: Dscp Based
      match dscp default policy custom
        priority 2 packet-loss-rate threshold 5.0 percent
        priority 1 one-way-delay threshold 50 msec
        priority 3 jitter threshold 200000 usec
        priority 2 byte-loss-rate threshold 5.0 percent
        Number of Traffic classes using this policy: 1

  class default
      match dscp all

------------------------------------------------------------------------------------------------

HubMC2# show domain default master discovered

------------------------------------------------------------------------------------------------
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 14:31:58.410 CET Tue Mar 17 2015


  *** Domain MC DISCOVERED sites ***

  Number of sites:  5
 *Traffic classes [Performance based][Load-balance based]

 Site ID: 255.255.255.255
  Site Discovered:06:32:33 ago
    Off-limits: Disabled
    DSCP :default[0]-Number of traffic classes[0][0]
    DSCP :af31[26]-Number of traffic classes[0][0]

 Site ID: 10.8.3.3
  Site Discovered:06:30:37 ago
    Off-limits: Disabled
    DSCP :default[0]-Number of traffic classes[0][0]
    DSCP :af31[26]-Number of traffic classes[0][0]

 Site ID: 10.2.10.10
  Site Discovered:06:30:37 ago
    Off-limits: Disabled
```

```
    DSCP :default[0]-Number of traffic classes[1][0]
    DSCP :af31[26]-Number of traffic classes[1][0]

Site ID: 10.2.11.11
  Site Discovered:06:30:34 ago
    Off-limits: Disabled
    DSCP :default[0]-Number of traffic classes[0][0]
    DSCP :af31[26]-Number of traffic classes[0][0]

 Site ID: 10.2.12.12
  Site Discovered:06:30:37 ago
    Off-limits: Disabled
    DSCP :default[0]-Number of traffic classes[0][0]
    DSCP :af31[26]-Number of traffic classes[0][0]
-------------------------------------------------------------------------------------------------

BR94# show domain default border status

-------------------------------------------------------------------------------------------------

**** Border Status ****

Instance Status: UP
Present status last updated: 06:39:21 ago
Loopback: Configured Loopback0 UP (10.9.4.4)
Master: 10.9.3.3
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 06:39:15
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Minimum Requirement: Met
External Wan interfaces:
     Name: Tunnel100 Interface Index: 11 SNMP Index: 8 SP: MPLS path-id: 30 Status: DOWN
Zero-SLA: NO

Auto Tunnel information:

   Name:Tunnel0 if_index: 12
   Borders reachable via this tunnel:  10.9.5.5

-------------------------------------------------------------------------------------------------

BR94# show domain default border site-prefix

-------------------------------------------------------------------------------------------------

Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id              Site-prefix          Last Updated         DC Bitmap  Flag
-------------------------------------------------------------------------------
10.2.10.10            10.1.10.0/24         00:36:58 ago         0x0         S
10.2.11.11            10.1.11.0/24         00:37:02 ago         0x0         S
10.2.10.10            10.2.10.10/32        00:36:58 ago         0x0         S
10.2.11.11            10.2.11.11/32        00:37:02 ago         0x0         S
10.2.12.12            10.2.12.12/32        00:40:37 ago         0x0         S
10.8.3.3              10.8.3.3/32          00:40:29 ago         0x1         S
10.9.3.3              10.8.0.0/16          00:38:40 ago         0x5         S,C,M
10.8.3.3              10.8.0.0/16          00:38:40 ago         0x5         S,C,M
10.9.3.3              10.9.3.3/32          00:38:40 ago         0x4         S
10.9.3.3              10.9.0.0/16          00:38:40 ago         0x5         S,C,M
10.8.3.3              10.9.0.0/16          00:38:40 ago         0x5         S,C,M
```

```
255.255.255.255      *10.0.0.0/8           00:40:29 ago         0x1         S,T
------------------------------------------------------------------------------------------------

R10# show domain default master channels dst-site-id 10.8.3.3

------------------------------------------------------------------------------------------------

Legend: * (Value obtained from Network delay:)

Channel Id: 27  Dst Site-Id: 10.8.3.3  Link Name: INET  DSCP: default [0] pfr-label: 0:20
| 0:0 [0x140000] TCs: 0
  Channel Created: 01:16:34 ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Interface Id: 12
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 5 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    10.8.3.3/32 (Active)
    10.8.0.0/16 (Active)
    10.9.0.0/16 (Standby)
  ODE Stats Bucket Number: 1
   Last Updated  : 00:00:24 ago
    Packet Count  : 562
Byte Count    : 47208
    One Way Delay : 71 msec*
    Loss Rate Pkts: 0.0 %
    Loss Rate Byte: 0.0 %
    Jitter Mean   : 619 usec
    Unreachable   : FALSE
  ODE Stats Bucket Number: 2
   Last Updated  : 00:00:54 ago
    Packet Count  : 558
    Byte Count    : 46872
    One Way Delay : 55 msec*
    Loss Rate Pkts: 0.0 %
    Loss Rate Byte: 0.0 %
    Jitter Mean   : 556 usec
    Unreachable   : FALSE
   TCA Statistics:
      Received:133 ; Processed:133 ; Unreach_rcvd:0
  Latest TCA Bucket
   Last Updated  : 00:00:24 ago
    One Way Delay : 71 msec*
    Loss Rate Pkts: NA
    Loss Rate Byte: NA
    Jitter Mean   : NA
    Unreachability: FALSE
.
.
.
------------------------------------------------------------------------------------------------

R10# show domain default border status

------------------------------------------------------------------------------------------------

Tue Mar 24 04:52:50.379

**** Border Status ****
```

```
Instance Status: UP
Present status last updated: 3d14h ago
Loopback: Configured Loopback0 UP (10.2.10.10)
Master: 10.2.10.10
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 3d14h
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Minimum Requirement: Met
External Wan interfaces:
Name: Tunnel100 Interface Index: 14 SNMP Index: 8 SP: MPLS Status: UP Zero-SLA: NO Path-id
 List: 0:10, 1:30
Name: Tunnel200 Interface Index: 15 SNMP Index: 9 SP: INET Status: UP Zero-SLA: NO Path-id
 List: 0:20, 1:40

Auto Tunnel information:

Name:Tunnel0 if_index: 13
Borders reachable via this tunnel:

-------------------------------------------------------------------------------------------------

R10# show domain default master status

-------------------------------------------------------------------------------------------------

*** Domain MC Status ***

Master VRF: Global

Instance Type: Branch
Instance id: 0
Operational status: Up
Configured status: Up
Loopback IP Address: 10.2.10.10
Load Balancing:
Operational Status: Up
Max Calculated Utilization Variance: 1%
Last load balance attempt: never
Last Reason: Variance less than 20%
Total unbalanced bandwidth:
External links: 0 Kbps Internet links: 0 Kbps
Route Control: Enabled
Mitigation mode Aggressive: Disabled
Policy threshold variance: 20
Minimum Mask Length: 28
Minimum Requirement: Met

Borders:
IP address: 10.2.10.10
Version: 2
Connection status: CONNECTED (Last Updated 3d14h ago )
Interfaces configured:
Name: Tunnel100 | type: external | Service Provider: MPLS | Status: UP | Zero-SLA: NO
Number of default Channels: 0

Path-id list: 0:10 1:30

Name: Tunnel200 | type: external | Service Provider: INET | Status: UP | Zero-SLA: NO
Number of default Channels: 0
```

```
Path-id list: 0:20 1:40

Tunnel if: Tunnel0

------------------------------------------------------------------------------------------------

R10# show domain default master site-capability 10.9.3.3 path-id

------------------------------------------------------------------------------------------------

Site id : 10.9.3.3
Site pop id : 1
Site mc type : Transit
Border Address : 10.9.4.4
Service provider: MPLS path-id: 30 if_index: 11
Border Address : 10.9.5.5
Service provider: INET path-id: 40 if_index: 11

------------------------------------------------------------------------------------------------

R10# show domain default master site-capability 10.8.3.3 path-id

------------------------------------------------------------------------------------------------

Site id : 10.8.3.3
Site pop id : 0
Site mc type : Hub
Border Address : 10.8.5.5
Service provider: INET path-id: 20 if_index: 11
Border Address : 10.8.4.4
Service provider: MPLS path-id: 10 if_index: 11

------------------------------------------------------------------------------------------------

R10#  show domain default border channels service-provider INET

------------------------------------------------------------------------------------------------

Tue Mar 24 04:53:39.968

Border Smart Probe Stats:

Smart probe parameters:
Source address used in the Probe: 10.2.10.10
Unreach time: 1000 ms
Probe source port: 18000
Probe destination port: 19000
Interface Discovery: ON
Probe freq for channels with traffic :10 secs
Discovery Probes: OFF
Number of transit probes consumed :0
Number of transit probes re-routed: 0
DSCP's using this: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]
[16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33]
[34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51]
[52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64]
All the other DSCPs use the default interval: 10 secs


Channel id: 6
Channel create time: 3d14h ago
Site id : 10.8.3.3
DSCP : default[0]
Service provider : INET
Pfr-Label : 0:20 | 0:0 [0x140000]
```

```
exit path-id: 0
Exit path-id sent on wire: 0
Number of Probes sent : 5657983
Number of Probes received : 5823008
Last Probe sent : 00:00:00 ago
Last Probe received : 00:00:00 ago
Channel state : Discovered and open
Channel next_hop : 10.0.200.85
RX Reachability : Reachable
TX Reachability : Reachable
Channel is sampling 0 flows
Channel remote end point: 10.0.200.85
Channel to hub: TRUE
Version: 3
Supports Zero-SLA: Yes
Muted by Zero-SLA: No
Probe freq with traffic : 1 in 10000 ms
.
.
.

------------------------------------------------------------------------------------------------

R10#  show ip nhrp nhs
------------------------------------------------------------------------------------------------

Legend: E=Expecting replies, R=Responding, W=Waiting
Tunnel100:
10.0.100.84 RE NBMA Address: 172.16.84.4 priority = 0 cluster = 0
10.0.100.94 RE NBMA Address: 172.16.94.4 priority = 0 cluster = 0

Tunnel200:
10.0.200.85 RE NBMA Address: 172.16.85.5 priority = 0 cluster = 0
10.0.200.95 RE NBMA Address: 172.16.95.5 priority = 0 cluster = 0


------------------------------------------------------------------------------------------------
```

CHAPTER 278

PfRv3 Zero SLA Support

The Performance Routing v3 (PfRv3) Zero SLA Support feature enables users to reduce probing frequency on various ISP links, such as 3G, 4G, and LTE. When the Zero SLA (0-SLA) feature is configured on an ISP link, only the channel with the DSCP (Differentiated Services Code Point) value 0 is probed. For all other DSCPs, channels are created only if there is traffic, but no probing is performed.

Feature Information for PfRv3 Zero SLA Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 334: Feature Information for PfRv3 Zero SLA Support

Feature Name	Releases	Feature Information
PfRv3 Path of Last Resort Support	15.5(3)M	The PfRv3 Path of Last Resort is a route used by the device when a service provider cannot be reached or the exits are out of bandwidth. The following commands were modified or added by this feature: **domain path isp-name**, **show domain default vrf border**, **show domain default vrf master**.

Feature Name	Releases	Feature Information
Performance Routing v3 Zero SLA Support	15.5(1)T Cisco IOS XE Release 3.14S	The Performance Routing v3 Zero SLA Support enables users to reduce probing frequency on various ISP links. The following command was modified by this feature:**domain** (interface configuration).

Prerequisites for PfRv3 Zero SLA Support

- Upgrade hub-border routers with the latest Cisco IOS image to configure the Zero SLA feature.

Restrictions for PfRv3 Zero SLA Support

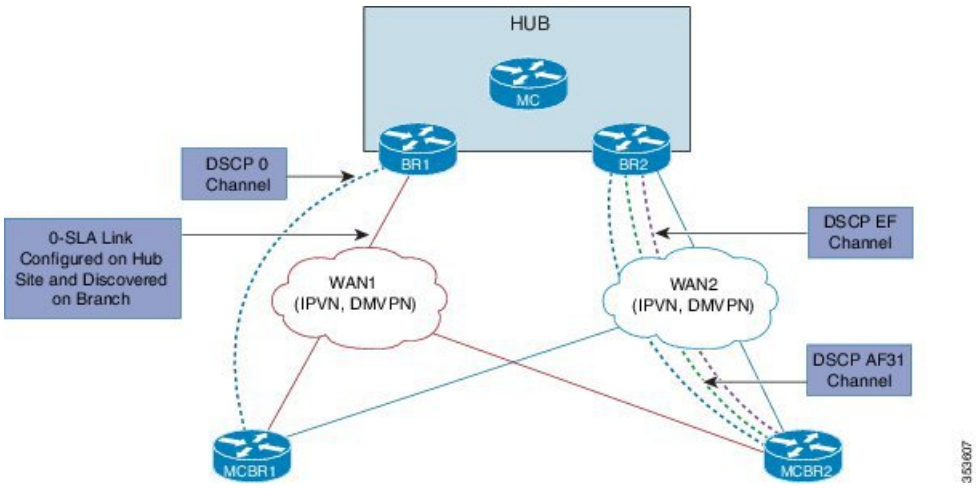
- Fast-monitor interval and brown out features are not supported with Zero SLA configurations.

Information About PfRv3 Zero SLA Support

Information About Zero SLA

The Zero SLA (0-SLA) feature enables users to reduce probing frequency in their network infrastructure. Reduction in probing process helps in reducing cost especially when ISPs charge based on traffic, and helps in optimizing network performance when ISPs provide limited bandwidth. When this feature is configured, probe is sent only on the DSCP-0 channel. For all other DSCPs, channels are created if there is traffic, but no probing is performed. The reachability of other channels is learnt from the DSCP-0 channel that is available at the same branch site.

Figure 232: Probing on Zero SLA



In the above illustration, the branch and hub sites are connected with red and blue ISP links. On the red ISP link, Zero SLA is configured at the hub site. Traffic exists on DSCP-0, DSCP AF31, and DSCP-EF channels on both ISP links, but on the red link probing is sent only on the DSCP-0 channel. A probe sent during the WAN discovery signals if a link is a Zero SLA link or a normal link.

Information About Path of Last Resort

A Path of Last Resort is a route used by the device when a service provider cannot be reached or the exits are out of bandwidth. This feature is supported for 3G and 4G metered links. When the service provider is not available, the traffic is routed to the path of last resort if you have specified the **path of last resort** keyword in the **domain path** command. When the exits are up with optimum bandwidth, the links are transitioned back. The following are the different supported modes:

- Standby mode—No traffic classes are routed over the path of last resort service provider.
- Active mode—Traffic classes are routed over the path of last resort service provider.
- Disabled mode—The path of last resort is not enabled for the interface.

The path of last resort routes are muted when it is in standby mode. The smart probe frequency is reduced to 1 packet every 10 seconds from 20 packets per second.

Compatibility Matrix for Zero SLA Support

In Performance Routing v3, capability negation happens through service advertisement framework (SAF) messages. When the PfR v3 domain comes up, it registers itself to the SAF to publish the compatibility and support for different release versions.

Use the **show domain default master site-capability** command to view the release version and the capability negation between hub and branch sites.

The following table shows the devices with various Cisco IOS/XE release versions and its support for Zero SLA within a single branch.

Master Controller	Border Router	Compatibility Between Release Versions	Zero SLA Support
Cisco IOS XE Release 3.14 or later Cisco IOS Release 15.5(1)T or later	Cisco IOS XE Release 3.14 or later Cisco IOS Release 15.5(1)T or later	Yes	If the master controller and border routers have the same Cisco IOS release versions, the Zero SLA feature is enabled.
Cisco IOS XE Release 3.14 or later Cisco IOS Release 15.5(1)T or later	Cisco IOS XE Release 3.13 or earlier Cisco IOS Release 15.4T or earlier	Yes	If the master controller has the latest Cisco IOS release and the border router has the earlier release version, the Zero SLA feature is disabled.
Cisco IOS XE Release 3.13 Cisco IOS Release 15.4T	Cisco IOS XE Release 3.13 Cisco IOS Release 15.4T	Yes	Zero SLA is not supported on Cisco IOS XE Release 3.13.
Cisco IOS XE Release 3.13 Cisco IOS Release 15.4T	Cisco IOS XE Release 3.14 or later Cisco IOS Release 15.5(1)T or later	No	The release versions are not compatible and hence, Zero SLA cannot be enabled.

Note

- If you are configuring PfRv3 on a site, it is mandatory that the hub master and the hub border routers in a site are on the same version of the Cisco IOS XE software.
- In a site, the branch master controller and the associated borders in the branch should also have the same version of the Cisco IOS XE software. But, it is not mandatory for the software version on the hub master or the hub border to match the software version on the branch master controller and its borders.
- Ensure that the Cisco IOS XE software version installed on the hub master, hub border router, branch master controller and borders support Zero SLA.

How to Configure PfRv3 Zero SLA Support

Configuring PfRv3 Zero SLA Support

Configure the Zero SLA (0-SLA) feature on the border router at the hub site.

Before you begin

Configure PfRv3 topology on the hub and branch site. For more information on configuring PfRv3, see the "How to Configure PfRv3" topic in the *Performance Routing v3 Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **bandwidth** *bandwidth-value*
5. **ip address** *ip-address mask*
6. **domain path** *isp-name* [**internet-bound** | **path-id** | **path-last-resort** | **zero-sla**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 100`	Enters interface configuration mode.
Step 4	**bandwidth** *bandwidth-value* **Example:** `Device(config-if)# bandwidth 10000000`	Configures inherited and received bandwidth values for the tunnel interface. The bandwidth value is in kilobits and the valid values are 1 to 10000000.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.32.1.1 255.0.0.0`	Configures an IP address of the border router at the hub site.
Step 6	**domain path** *isp-name* [**internet-bound** | **path-id** | **path-last-resort** | **zero-sla**] **Example:** `Device(config-if)# domain path ISP1 zero-sla`	Specifies a service provider for the interface. • **internet-bound**—Configures an internet bound interface. • **path-id**—Configures service provider's path-id for the interface. • **path-last-resort**—Configures the interface to be a path of a last resort. • **zero-sla**—Configures Zero SLA for the interface. **Note** You can configure multiple Internet Service Providers (ISPs). If you are defining a specific domain name for an ISP (for example, domain_abc), you must specify the same domain name while configuring the ISP paths.

Verifying PfRv3 Zero SLA Support

The **show** commands can be entered in any order.

Before you begin

Configure Zero SLA on the hub-border router.

SUMMARY STEPS

1. **show domain default master status**
2. **show domain default master channel**
3. **show domain default border status**
4. **show domain default border channel**
5. **show domain default master site-capability**
6. **show domain default vrf** *vrf-name* **master status**
7. **show domain default vrf** *vrf-name* **border status**
8. **show domain default vrf** *vrf-name* **master channels**
9. **show domain default vrf** *vrf-name* **border channels**
10. **show domain default vrf** *vrf-name* **master policy**

DETAILED STEPS

Step 1 **show domain default master status**

Displays the status of the hub master controller.

Step 2 **show domain default master channel**

Displays channel information of the hub master controller.

Step 3 **show domain default border status**

Displays the status of the hub border routers.

Step 4 **show domain default border channel**

Displays the information of border router channels at the hub site.

Step 5 **show domain default master site-capability**

Displays the capability information of master controller.

Example:

```
Device# show domain default master site-capability

Device Capability

------------------------------------------------------------
|       Capability          |     Major     |     Minor      |
------------------------------------------------------------
|     Domain                |     2         |      0         |
------------------------------------------------------------
|     Zero-SLA              |     1         |      0         |
------------------------------------------------------------
```

```
Site id  :10.2.10.10

------------------------------------------------------------
|       Capability          |     Major     |     Minor      |
------------------------------------------------------------
|     Domain                |     2         |      0         |
------------------------------------------------------------
|     Zero-SLA              |     1         |      0         |
------------------------------------------------------------
Site id  :10.2.12.12

------------------------------------------------------------
|     Capability            |     Major     |     Minor      |
------------------------------------------------------------
|     Domain                |     2         |      0         |
------------------------------------------------------------
|     Zero-SLA              |     1         |      0         |
------------------------------------------------------------
```

Table 335: ***show domain default master site-capability*** *Field Descriptions*

Field	Description
Capability	Features supported by PfR v3 domain.
Domain	Domain version. Major - Means the major release version number for PfR v3. Minor - Means the minor release version number for PfR v3.
Zero-SLA	Zero-SLA feature support. Major - Means the major release version of the Zero-SLA feature on the master controller. Minor - Means the minor release version of the Zero-SLA feature on the master controller.

Step 6 **show domain default vrf** *vrf-name* **master status**

Displays the master status of the hub border routers.

Example:

```
Device# show domain default vrf vrf1 master status

Borders:
  IP address: 10.204.1.4
  Version: 2
  Connection status: CONNECTED (Last Updated 00:59:16 ago )
  Interfaces configured:
   Name: Tunnel20 | type: external | Service Provider: ISP2 | Status: UP | Zero-SLA: NO | Path of
Last Resort: Disabled
    Number of default Channels: 0
  Tunnel if: Tunnel1
  IP address: 10.203.1.3
  Version: 2
  Connection status: CONNECTED (Last Updated 00:59:16 ago )
```

```
  Interfaces configured:
    Name: Tunnel10 | type: external | Service Provider: ISP1 | Status: UP | Zero-SLA: YES | Path
of
Last Resort: Standby
      Number of default Channels: 0
  Tunnel if: Tunnel1
```

Step 7 **show domain default vrf** *vrf-name* **border status**

Displays the master status of the hub border routers.

Example:

```
Device# show domain default vrf vrf1 border status

------------------------------------------------------------------------
**** Border Status ****
Instance Status: UP
Present status last updated: 01:01:42 ago
Loopback: Configured Loopback1 UP (30.209.1.9)
Master: 30.209.1.9
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 01:01:42
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Minimum Requirement: Met
External Wan interfaces:
    Name: Tunnel10 Interface Index: 16 SNMP Index: 13 SP: ISP1 path-id: 0 Status: UP Zero-SLA: YES
Path of Last Resort: Standby Path-id List: 0:0
  Name: Tunnel20 Interface Index: 18 SNMP Index: 15 SP: ISP2 Status: UP Zero-SLA: NO Path of Last
Resort: Disabled Path-id List: 0:0


Auto Tunnel information:


   Name:Tunnel1 if_index: 21
   Borders reachable via this tunnel:
------------------------------------------------------------------------
```

Step 8 **show domain default vrf** *vrf-name* **master channels**

Displays the master status of the hub master controller.

Example:

```
Device# show domain default vrf vrf1 master channels

Channel Id: 9 Dst Site-Id: 30.209.1.9 Link Name: ISP1 DSCP: af41 [34] pfr-label: 0:0 | 0:0 [0x0]
TCs: 0
  Channel Created: 00:57:15 ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: FALSE
  Interface Id: 16
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: Yes
  Muted by Path of Last Resort: Yes
  Estimated Channel Egress Bandwidth: 0 Kbps
  Immitigable Events Summary:
    Total Performance Count: 0, Total BW Count: 0
```

```
  ODE Stats Bucket Number: 1
    Last Updated : 00:56:15 ago
     Packet Count : 505
     Byte Count : 42420
     One Way Delay : 229 msec*
     Loss Rate Pkts: 0.0 %
     Loss Rate Byte: 0.0 %
     Jitter Mean : 535 usec
     Unreachable : FALSE
    TCA Statistics:
     Received:1 ; Processed:1 ; Unreach_rcvd:0
  Latest TCA Bucket
    Last Updated : 00:56:15 ago
     One Way Delay : 229 msec*
     Loss Rate Pkts: NA
     Loss Rate Byte: NA
     Jitter Mean : NA
     Unreachability: FALSE
```

Step 9 **show domain default vrf** *vrf-name* **border channels**

Displays the information of border router channels at the hub site.

Example:

```
Device# show domain default vrf vrf1 border channels

Channel id: 2
 Channel create time: 00:46:02 ago
 Site id : 255.255.255.255
 DSCP : default[0]
 Service provider : ISP1
 Pfr-Label : 0:0 | 0:0 [0x0]
 exit path-id: 0
 Exit path-id sent on wire: 0
 Number of Probes sent : 0
 Number of Probes received : 0
 Last Probe sent : 00:46:02 ago
 Last Probe received : - ago
 Channel state : Initiated and open
 Channel next_hop : 0.0.0.0
 RX Reachability : Initial State
 TX Reachability : Reachable
 Channel is sampling 0 flows
 Channel remote end point: 0.0.0.0
 Channel to hub: FALSE
 Version: 0
 Supports Zero-SLA: No
 Muted by Zero-SLA: No
 Muted by Path of Last Resort: Yes
 Probe freq with traffic : 1 in 10000 ms
```

Step 10 **show domain default vrf** *vrf-name* **master policy**

Displays the status of the master policy.

Example:

```
Device# show domain default vrf vrf1 master policy

class VOICE sequence 10
  path-last-resort ISP1
  class type: Dscp Based
    match dscp ef policy custom
```

```
priority 1 one-way-delay threshold 200 msec
Number of Traffic classes using this policy: 2
```

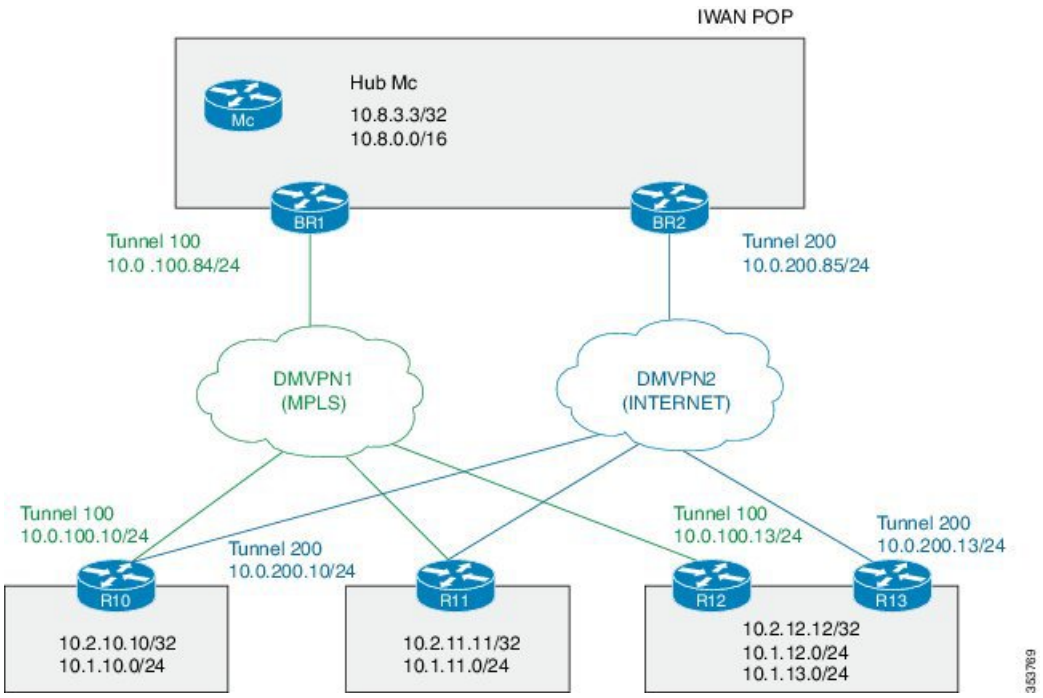
Configuration Examples for PfRv3 Zero SLA Support

Example: Configuring PfRv3 Zero SLA Support

Let us consider a use case scenario, where the service provider of a large enterprise network wants to reduce the probing frequency on all its channels. To reduce probing, Zero-SLA is configured on the ISP link from BR1.

Note In the following example, only the hub master controller, BR1 (border router 1), R10 and R11 (branch border router) configurations are described.

Figure 233: PfRv3 Topology



In this example, the following routers are used:

- Hub Master Controller — Cisco ASR 1002-X router configured with an embedded services processor (ESP) default bandwidth of 5 Gbps upgradable with software licensing options to 10 Gbps, 20 Gbps, and 36 Gbps.
- Hub Border Routers — Cisco ASR 1000 Series Embedded Services Processor 2

- Branch Routers — Cisco 4451X Integrated Services Router.

Configure the interfaces on hub master controller

```
HubMC> enable
HubMC# configure terminal
HubMC(config)# interface Loopback0
HubMC(config-if)# ip address 10.8.3.3 255.255.255.255
HubMC(config-if)# exit
```

Configure the device as hub-master controller

```
HubMC(config)# domain one
HubMC(config-domain)# vrf default
HubMC(config-domain-vrf)# master hub
HubMC(config-domain-vrf-mc)# source-interface Loopback0
HubMC(config-domain-vrf-mc)# enterprise-prefix prefix-list ENTERPRISE
HubMC(config-domain-vrf-mc)# site-prefixes prefix-list DATA_CENTER_1
HubMC(config-domain-vrf-mc)# exit
```

Configure IP prefix-lists

```
HubMC(config)# ip prefix-list DATA_CENTER_1 seq 5 permit 10.8.0.0/16 le 24
HubMC(config)# ip prefix-list ENTERPRISE seq 5 permit 10.0.0.0/8 le 24
```

Configure domain policies on hub master controller

```
HubMC(config)# domain one
HubMC(config-domain)# vrf default
HubMC(config-domain-vrf)# master hub
HubMC(config-domain-vrf-mc)# monitor-interval 2 dscp ef
HubMC(config-domain-vrf-mc)# load-balance
HubMC(config-domain-vrf-mc)# class VOICE sequence 10
HubMC(config-domain-vrf-mc-class)# match dscp ef policy voice
HubMC(config-domain-vrf-mc-class)# path-preference MPLS fallback INET
HubMC(config-domain-vrf-mc-class)# exit
HubMC(config-domain-vrf-mc)# class VIDEO sequence 20
HubMC(config-domain-vrf-mc-class)# match dscp af41 policy real-time-video
HubMC(config-domain-vrf-mc-class)# match dscp cs4 policy real-time-video
HubMC(config-domain-vrf-mc-class)# path-preference INET fallback MPLS
HubMC(config-domain-vrf-mc-class)# exit
HubMC(config-domain-vrf-mc)# class CRITICAL sequence 30
HubMC(config-domain-vrf-mc-class)# match dscp af31 policy custom
HubMC(config-domain-vrf-mc-class-type)# priority 2 loss threshold 10
HubMC(config-domain-vrf-mc-class-type)# priority 1 one-way-delay threshold 600
HubMC(config-domain-vrf-mc-class-type)# priority 2 jitter threshold 600
HubMC(config-domain-vrf-mc-class)# exit
HubMC(config-domain-vrf-mc-class)# path-preference MPLS fallback INET
```

Configure the interfaces on hub border router (BR1)

```
BR1> enable
BR1# configure terminal
BR1(config)# interface Loopback0
BR1(config-if)# ip address 10.8.1.1 255.255.255.255
BR1(config-if)exit
```

Configure the device as border router (BR1)

```
BR1(config)# domain one
BR1(config-domain)# vrf default
BR1(config-domain-vrf)# border
BR1(config-domain-vrf-br)# source-interface Loopback0
BR1(config-domain-vrf-br)# master 10.8.3.3
BR1(config-domain-vrf-br)# exit
```

Configure tunnel from BR1 to DMVPN1 (MPLS)Link

```
BR1(config)# interface Tunnel100
BR1(config-if)# bandwidth 100000
BR1(config-if)# ip address 10.0.100.84 255.255.255.0
BR1(config-if)# no ip redirects
BR1(config-if)# ip mtu 1400
BR1(config-if)# ip nhrp authentication cisco
BR1(config-if)# ip nhrp map multicast dynamic
BR1(config-if)# ip nhrp network-id 1
BR1(config-if)# ip nhrp holdtime 600
BR1(config-if)# ip tcp adjust-mss 1360
BR1(config-if)# load-interval 30
BR1(config-if)# tunnel source GigabitEthernet3
BR1(config-if)# tunnel mode gre multipoint
BR1(config-if)# tunnel key 100
BR1(config-if)# tunnel protection ipsec profile DMVPN-PROFILE1
BR1(config-if)# domain one path MPLS
```

Configure Zero-SLA on BR1 to DMVPN1 (MPLS)Link

```
BR1(config-if)# domain one path MPLS zero-sla
```

Configure the interfaces (R10)

```
R10> enable
R10# configure terminal
R10(config)# interface Loopback0
R10(config-if)# ip address 10.2.10.10 255.255.255.255
R10(config-if)exit
```

Configure the device as branch master controller (R10)

```
R10(config)# domain one
R10(config-domain)# vrf default
R10(config-domain-vrf)# border
R10(config-domain-vrf-br)# source-interface Loopback0
R10(config-domain-vrf-br)# master local
R10(config-domain-vrf-br)# exit
R10(config-domain-vrf)# master branch
R10(config-domain-vrf-mc)# source-interface Loopback0
R10(config-domain-vrf-mc)# hub 10.8.3.3
```

Configure the tunnel interface and tunnel path from R10

```
R10(config)# interface Tunnel100
R10(config-if)# bandwidth 100000
R10(config-if)# ip address 10.0.100.10 255.255.255.0
R10(config-if)# no ip redirects
R10(config-if)# ip mtu 1400
R10(config-if)# ip nhrp authentication cisco
R10(config-if)# ip nhrp map 10.0.100.84 172.16.84.4
R10(config-if)# ip nhrp map multicast 172.16.84.4
R10(config-if)# ip nhrp network-id 1
R10(config-if)# ip nhrp holdtime 600
R10(config-if)# ip nhrp nhs 10.0.100.84
R10(config-if)# ip nhrp registration timeout 60
R10(config-if)# ip tcp adjust-mss 1360
R10(config-if)# load-interval 30
R10(config-if)# delay 1000
R10(config-if)# tunnel source GigabitEthernet2
R10(config-if)# tunnel mode gre multipoint
R10(config-if)# tunnel key 100
R10(config-if)# tunnel protection ipsec profile DMVPN-PROFILE1
R10(config-if)# domain one path MPLS
```

Configure another tunnel path from R10

```
R10(config)# interface Tunnel200
R10(config-if)# bandwidth 50000
R10(config-if)# ip address 10.0.200.10 255.255.255.0
R10(config-if)# no ip redirects
R10(config-if)# ip mtu 1400
R10(config-if)# ip nhrp authentication cisco
R10(config-if)# ip nhrp map 10.0.200.85 172.16.85.5
R10(config-if)# ip nhrp multicast 172.16.85.5
R10(config-if)# ip nhrp network-id 2
R10(config-if)# ip nhrp holdtime 600
R10(config-if)# ip nhrp nhs 10.0.200.85
R10(config-if)# ip tcp adjust-mss 1360
R10(config-if)# load-interval 30
R10(config-if)# delay 1000
R10(config-if)# tunnel source GigabitEthernet3
R10(config-if)# tunnel mode gre multipoint
R10(config-if)# tunnel key 200
R10(config-if)# tunnel protection ipsec profile DMVPN-PROFILE2
R10(config-if)# domain one path INET
```

Configure the interfaces (R11)

```
R11> enable
R11# configure terminal
R11(config)# interface Loopback0
R11(config-if)# ip address 10.2.11.11 255.255.255.255
R11(config-if)# exit
```

Configure the device as branch master controller (R11)

```
R11(config)# domain one
R11(config-domain)# vrf default
R11(config-domain-vrf)# border
R11(config-domain-vrf-br)# source-interface Loopback0
R11(config-domain-vrf-br)# master local
R11(config-domain-vrf-br)# exit
R11(config-domain-vrf)# master branch
R11(config-domain-vrf-mc)# source-interface Loopback0
R11(config-domain-vrf-mc)# hub 10.8.3.3
```

Configure the tunnel interface and tunnel path from R11

```
R11(config)# interface Tunnel100
R11(config-if)# bandwidth 100000
R11(config-if)# ip address 10.0.100.11 255.255.255.0
R11(config-if)# no ip redirects
R11(config-if)# ip mtu 1400
R11(config-if)# ip nhrp authentication cisco
R11(config-if)# ip nhrp map 10.0.100.84 172.16.84.4
R11(config-if)# ip nhrp map multicast 172.16.84.4
R11(config-if)# ip nhrp network-id 1
R11(config-if)# ip nhrp holdtime 600
R11(config-if)# ip nhrp nhs 10.0.100.84
R11(config-if)# ip nhrp registration timeout 60
R11(config-if)# ip tcp adjust-mss 1360
R11(config-if)# load-interval 30
R11(config-if)# delay 1000
R11(config-if)# tunnel source GigabitEthernet2
R11(config-if)# tunnel mode gre multipoint
R11(config-if)# tunnel key 100
R11(config-if)# tunnel protection ipsec profile DMVPN-PROFILE1
R11(config-if)# domain one path MPLS
```

Configure another tunnel path from R11

```
R11(config)# interface Tunnel200
R11(config-if)# bandwidth 50000
R11(config-if)# ip address 10.0.200.11 255.255.255.0
R11(config-if)# no ip redirects
R11(config-if)# ip mtu 1400
R11(config-if)# ip nhrp authentication cisco
R11(config-if)# ip nhrp map 10.0.200.85 172.16.85.5
R11(config-if)# ip nhrp multicast 172.16.85.5
R11(config-if)# ip nhrp network-id 2
R11(config-if)# ip nhrp holdtime 600
R11(config-if)# ip nhrp nhs 10.0.200.85
R11(config-if)# ip tcp adjust-mss 1360
R11(config-if)# load-interval 30
R11(config-if)# delay 1000
R11(config-if)# tunnel source GigabitEthernet3
R11(config-if)# tunnel mode gre multipoint
R11(config-if)# tunnel key 200
R11(config-if)# tunnel vrf INET2
R11(config-if)# tunnel protection ipsec profile DMVPN-PROFILE2
R11(config-if)# domain one path INET
```

Verifying PfRv3 Zero-SLA Configurations

To verify the PfRv3 Zero-SLA configuration, use the following show commands in any order:

- **show domain** *domain-name* **master status**
- **show domain** *domain-name* **border status**
- **show domain** *domain-name* **master channel**
- **show domain** *domain-name* **border channel**
- **show domain** *domain-name* **master site-capability**

CHAPTER 279

PfRv3 Path of Last Resort

The PfRv3 path of last resort feature allows the traffic to be routed to the path of last resort.

Feature Information for PfRv3 Path of Last Resort

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 336: Feature Information for PfRv3 Path of Last Resort

Feature Name	Releases	Feature Information
PfRv3 Path of Last Resort	15.5(3)M	The PfRv3 Path of Last Resort is a route used by the device when a service provider cannot be reached or the exits are out of bandwidth. The following commands were modified or added by this feature: **domain path isp-name**, **show domain default vrf border**, **show domain default vrf master**.

Restrictions for PfRv3 Path of Last Resort

- Path of last resort supports probing per interface and not per channel.
- Path of last resort is not supported on multi next hop interfaces.

Information About PfRv3 Path of Last Resort

PfRv3 Path of Last Resort

The PfRv3 Path of Last Resort feature provides the ability to designate a service provider as a path of last resort such that when the primary and fallback service providers become unavailable due to unreadability or out of bandwidth situations, traffic is routed over the path of last resort service provider. This feature is used for metered links where data is charged on a per-usage basis and is used when no other service providers are available.

The following are the different supported modes:

- Standby mode—No traffic classes are currently routed over the path of last resort service provider.
- Active mode—Traffic classes are currently routed over the path of last resort service provider.
- Disabled mode—The path of last resort is not enabled.

The channels of the path of last resort are inactive when it is in standby mode. Once the path of last resort is active, smart probes are sent only on DSCP 0 (Zero SLA) to conserve bandwidth. In addition, smart probe frequency is reduced to 1 packet every 10 seconds from 20 packets per seconds, unreachable detection are extended to 60 seconds.

How to Configure PfRv3 Path of Last Resort

Configuring Policy for Path of Last Resort

To configure policy for path of last resort, perform the steps below.

SUMMARY STEPS

1. **domain default**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**domain default**
	Example:
	`domain default`
	`  vrf default`
	`  master hub`
	`  class foo seq 1`
	`  match dscp ef policy voice`
	`  path-preference ISP1 fallback ISP2`
	`  path-last-resort ISP4`
		The keyword specifics that the traffic for this policy is routed over the path of last resort when the primary and fallback service providers are unavailable.

Configuring Path of Last Resort

To configure path of last resort, perform the steps below.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **domain path** *isp-name* [**internet-bound** | **path-id** | **path-last-resort** | **zero-sla**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 100`	Enters interface configuration mode.
Step 4	**domain path** *isp-name* [**internet-bound** | **path-id** | **path-last-resort** | **zero-sla**] **Example:** `Device(config-if)# domain path ISP1 path-last-resort`	Specifies a service provider for the interface. • **internet-bound**—Configures an internet bound interface. • **path-id**—Configures service provider's path-id for the interface. • **path-last-resort**—Configures the interface to be a path of a last resort. • **zero-sla**—Configures Zero SLA for the interface. **Note** You can configure multiple Internet Service Providers (ISPs). If you are defining a specific domain name for an ISP (for example, domain_abc), you must specify the same domain name while configuring the ISP paths.

Verifying PfRv3 Path of Last Resort

The **show** commands can be entered in any order.

SUMMARY STEPS

1. **show domain default vrf** *vrf-name* **master status**
2. **show domain default vrf** *vrf-name* **border status**
3. **show domain default vrf** *vrf-name* **master channels**
4. **show domain default vrf** *vrf-name* **border channels**
5. **show domain default vrf** *vrf-name* **master policy**

DETAILED STEPS

Step 1 show domain default vrf *vrf-name* master status

Displays the master status of the hub border routers.

Example:

```
Device# show domain default vrf vrf1 master status

Borders:
  IP address: 10.204.1.4
  Version: 2
  Connection status: CONNECTED (Last Updated 00:59:16 ago )
  Interfaces configured:
   Name: Tunnel20 | type: external | Service Provider: ISP2 | Status: UP | Zero-SLA: NO | Path of
Last Resort: Disabled
    Number of default Channels: 0
  Tunnel if: Tunnel1
  IP address: 10.203.1.3
  Version: 2
  Connection status: CONNECTED (Last Updated 00:59:16 ago )
  Interfaces configured:
    Name: Tunnel10 | type: external | Service Provider: ISP1 | Status: UP | Zero-SLA: YES | Path of
Last Resort: Standby
      Number of default Channels: 0
  Tunnel if: Tunnel1
```

Step 2 show domain default vrf *vrf-name* border status

Displays the master status of the hub border routers.

Example:

```
Device# show domain default vrf vrf1 border status

------------------------------------------------------------------------
**** Border Status ****
Instance Status: UP
Present status last updated: 01:01:42 ago
Loopback: Configured Loopback1 UP (30.209.1.9)
Master: 30.209.1.9
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 01:01:42
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Minimum Requirement: Met
External Wan interfaces:
    Name: Tunnel10 Interface Index: 16 SNMP Index: 13 SP: ISP1 path-id: 0 Status: UP Zero-SLA: YES
```

```
Path of Last Resort: Standby Path-id List: 0:0
  Name: Tunnel20 Interface Index: 18 SNMP Index: 15 SP: ISP2 Status: UP Zero-SLA: NO Path of Last
Resort: Disabled Path-id List: 0:0


Auto Tunnel information:


   Name:Tunnel1 if_index: 21
   Borders reachable via this tunnel:
---------------------------------------------------------------------
```

Step 3 **show domain default vrf** *vrf-name* **master channels**

Displays the master status of the hub master controller.

Example:

```
Device# show domain default vrf vrf1 master channels

Channel Id: 9 Dst Site-Id: 30.209.1.9 Link Name: ISP1 DSCP: af41 [34] pfr-label: 0:0 | 0:0 [0x0]
TCs: 0
  Channel Created: 00:57:15 ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: FALSE
  Interface Id: 16
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: Yes
  Muted by Path of Last Resort: Yes
  Estimated Channel Egress Bandwidth: 0 Kbps
  Immitigable Events Summary:
    Total Performance Count: 0, Total BW Count: 0
  ODE Stats Bucket Number: 1
    Last Updated : 00:56:15 ago
     Packet Count : 505
     Byte Count : 42420
     One Way Delay : 229 msec*
     Loss Rate Pkts: 0.0 %
     Loss Rate Byte: 0.0 %
     Jitter Mean : 535 usec
     Unreachable : FALSE
    TCA Statistics:
     Received:1 ; Processed:1 ; Unreach_rcvd:0
  Latest TCA Bucket
    Last Updated : 00:56:15 ago
     One Way Delay : 229 msec*
     Loss Rate Pkts: NA
     Loss Rate Byte: NA
     Jitter Mean : NA
     Unreachability: FALSE
```

Step 4 **show domain default vrf** *vrf-name* **border channels**

Displays the information of border router channels at the hub site.

Example:

```
Device# show domain default vrf vrf1 border channels

Channel id: 2
 Channel create time: 00:46:02 ago
 Site id : 255.255.255.255
 DSCP : default[0]
 Service provider : ISP1
```

```
Pfr-Label : 0:0 | 0:0 [0x0]
exit path-id: 0
Exit path-id sent on wire: 0
Number of Probes sent : 0
Number of Probes received : 0
Last Probe sent : 00:46:02 ago
Last Probe received : - ago
Channel state : Initiated and open
Channel next_hop : 0.0.0.0
RX Reachability : Initial State
TX Reachability : Reachable
Channel is sampling 0 flows
Channel remote end point: 0.0.0.0
Channel to hub: FALSE
Version: 0
Supports Zero-SLA: No
Muted by Zero-SLA: No
Muted by Path of Last Resort: Yes
Probe freq with traffic : 1 in 10000 ms
```

Step 5 **show domain default vrf** *vrf-name* **master policy**

Displays the status of the master policy.

Example:

```
Device# show domain default vrf vrf1 master policy

class VOICE sequence 10
  path-last-resort ISP1
  class type: Dscp Based
    match dscp ef policy custom
      priority 1 one-way-delay threshold 200 msec
      Number of Traffic classes using this policy: 2
```

CHAPTER 280

PfRv3 Fallback Timer

PfRv3 can move a specific traffic class (TC) from a primary, preferred path to a backup path to optimize performance. Use fallback-timer to set the time interval (called timeout) for the next re-evaluation of the primary path. Increasing the time interval causes PfRv3 to wait longer before reassessing. This can help to prevent excessive switching between the primary and secondary paths.

Feature Information for PfRv3 Fallback Timer

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 337: Feature Information for PfRv3 Fallback Timer

Feature Name	Releases	Feature Information
PfRv3 Fallback Timer	Cisco IOS XE Gibraltar 16.10.1	The PfRv3 Fallback Timer sets the re-evaluation interval for re-evaluating the primary path after a traffic class has been changed to a backup path. The following commands were modified or added by this feature: **fallback-time**, **show domain vrf master**.

Prerequisites for PfRv3 Fallback Timer

- Latest Cisco IOS XE image

Information About PfRv3 Fallback Timer

Overview of Fallback Timer

As part of its intelligent path selection, PfRv3 can move a specific traffic class (TC) from a primary, preferred path to a backup path to optimize performance. After changing the TC to a backup path, PfRv3 re-evaluates the primary path to determine when to return the TC to the primary path. The re-evaluation occurs in cycles of a specific period of time, and continues for as long as the traffic is not on the primary path.

In some situations, if the primary path alternates between meeting the performance requirements specified for the TC and not meeting the requirements, the TC may be switched excessively between the primary and backup paths. This "bouncing" between paths reduces the stability of the TC.

To prevent excessive switching between paths, you can increase the evaluation interval (called timeout) and apply a dampening algorithm.

Figure 234: Default Timeout

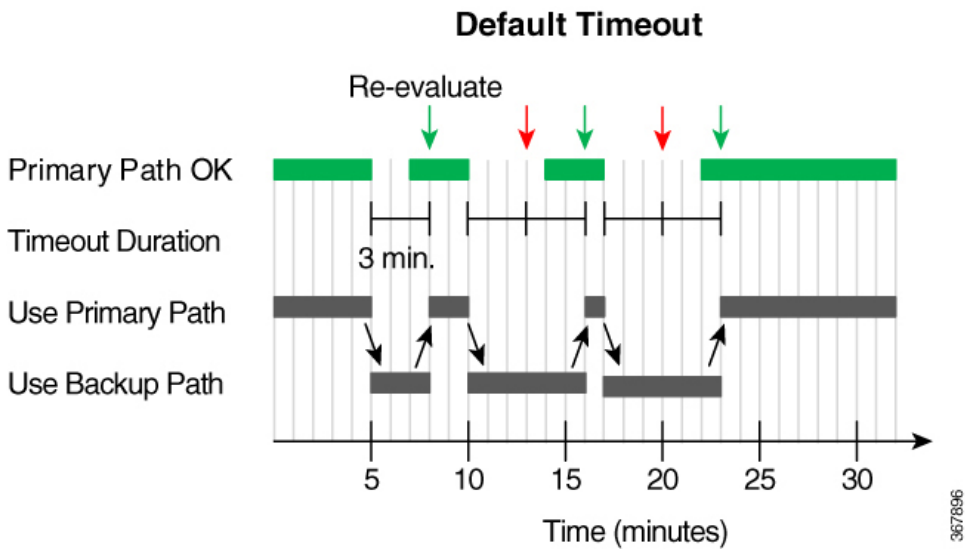
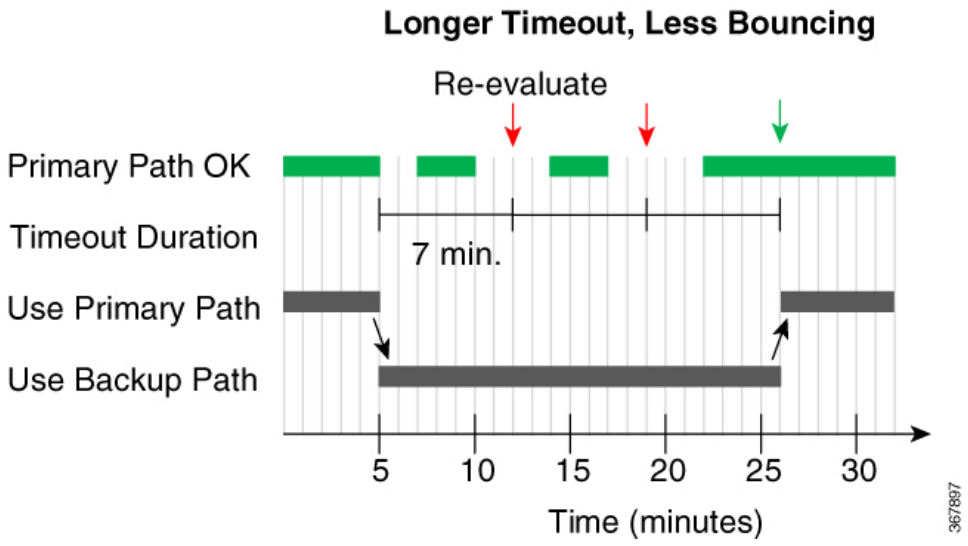


Figure 235: Longer Timeout Interval



Method	Description	Applicable to...
Increase evaluation interval	Use fallback-timer to set the time interval (called timeout) for the next re-evaluation of the primary path. Increasing the time interval causes PfRv3 to wait longer before reassessing. This prevents excessive switching between the primary and secondary paths. Possible values: 1 to 1440 minutes Default: 3 minutes	Global (per VRF) Traffic class
Dampening	Use fallback-timer to enable automatic adjustment of the re-evaluation time interval to prevent excessive switching between paths. When enabled, dampening temporarily increases the evaluation period if a traffic class has been switched more than once from the primary path to a backup path within a short time. It then gradually reduces the evaluation period over time if the primary path meets the performance requirements specified for the traffic class. Possible values: enable, disable Default: enable (if fallback-timer is configured)	Traffic class

How to Configure PfRv3 Fallback Timer

PfRv3 Fallback Timer Configuration

To configure the fallback timer, use:

fallback-timer *time-in-minutes* [**dampening** {**enable**/**disable**}]

See the examples below.

Fallback Timer Configuration Priority

Priority of fallback timer configuration:

per class (policy) > global > default

Example

Configuration:

- Traffic class A configuration: **fallback-timer 4 dampening enable**
- Global configuration: **fallback-timer 6**

Result:

Trafic class A will operate with a timeout of 4 minutes and dampening. Other traffic classes will have a fixed timeout (no dampening) of 6 minutes.

Viewing PfRv3 Fallback Timer Status

The **show** commands can be entered in any order.

Before you begin

Perform on hub master controller.

SUMMARY STEPS

1. **show domain** *domain-name* **vrf** *vrf-name* **master policy**
2. **show domain** *domain-name* **vrf** *vrf-name* **master traffic-classes detail**

DETAILED STEPS

Step 1 **show domain** *domain-name* **vrf** *vrf-name* **master policy**

Example

Sections of the output in bold are relevant to fallback timer.

```
Device# show domain default vrf green master policy
 No Policy publish pending
 Last publish Status : Peering Success
 Total publish errors : 0
------------------------------------------------------------------------------
Global-policy-list:

  class SER_CS1 sequence 10
    path-preference ISP1 fallback ISP2
    fallback timer timeout 5 minutes, dampening Enabled
    class type: Dscp Based
      match dscp cs1 policy custom
```

```
        priority 1 packet-loss-rate threshold 10.0 percent
        priority 1 byte-loss-rate threshold 10.0 percent
        Number of Traffic classes using this policy: 1

  class SER_EF sequence 20
    path-preference ISP1 fallback ISP2
    fallback timer timeout 6 minutes, dampening Disabled
    class type: Dscp Based
      match dscp ef policy custom
        priority 1 packet-loss-rate threshold 10.0 percent
        priority 1 byte-loss-rate threshold 10.0 percent
```

Step 2 **show domain** *domain-name* **vrf** *vrf-name* **master traffic-classes detail**

Example

Sections of the output in bold are relevant to fallback timer.

```
Device# show domain default vrf green master traffic-classes dscp cs1 detail

 Dst-Site-Prefix: 100.20.0.0/16       DSCP: cs1 [8] Traffic class id:28
  Clock Time:                 09:51:00 (CST) 08/24/2018
  TC Learned:                 00:10:29 ago
  Present State:              CONTROLLED
  Current Performance Status: in-policy
  Current Service Provider:   ISP2 path-id:4 since 00:01:50
  Previous Service Provider:  ISP1 pfr-label: 0:0 | 0:1 [0x1] for 488 sec
  (A fallback provider. Primary provider will be re-evaluated 00:03:11 later)
…
```

Configuration Examples for PfRv3 Fallback Timer

Example: Configuring PfRv3 Fallback Timer Globally

Configure the global fallback timer settings on a hub master controller.

Configure global fallback timer to 4 minutes

```
domain iwan
vrf default
master hub
advanced
fallback-timer 4
```

Disable fallback timer globally

Use **fallback-timer off** to disable re-evaluation of the primary path after a traffic class switches to a backup path. In this mode, traffic does not switch back to the primary path.

Note Consider restoring the fallback timer to the default 3 minutes instead of disabling.

```
domain iwan
vrf default
master hub
advanced
fallback-timer off
```

Example: Configuring PfRv3 Fallback Timer for Traffic Class

Configure the global fallback timer settings on a hub master controller.

Fallback 5 minutes, dampening enabled by default

```
domain iwan
vrf default
master hub
class VOICE sequence 10
match app audio policy voice
path-preference MPLS1 fallback INET1
fallback-timer 5
```

Fallback 10 minutes, dampening disabled

```
class REAL_TIME_VIDEO sequence 20
match dscp cs4 policy real-time-video
match dscp af41 policy real-time-video
path-preference MPLS1 fallback INET1
fallback-timer 10 dampening disable
```

Fallback timer off

Use **fallback-timer off** to disable re-evaluation of the primary path after a traffic class switches to a backup path. In this mode, traffic does not switch back to the primary path.

Note Consider restoring the fallback timer to the default 3 minutes instead of disabling.

```
class LOW_LATENCY_DATA sequence 30
match dscp cs2 policy real-time-video
match dscp af21 policy real-time-video
path-preference INET1 fallback MPLS1
fallback-timer off
```

CHAPTER 281

PfRv3 Probe Reduction

This document provides information about the PfRv3 Probe Reduction feature that allows reducing traffic probe on channels that do not carrying any traffic.

Prerequisites for PfRv3 Probe Reduction

Information About PfRv3 Probe Reduction

The PfRv3 Probe Reduction feature allows reducing traffic probe on channels that do not carry any traffic. Probing is used to compute important metrics such as reachability, one-way delay (OWD), jitter, and loss on channels that do not have user traffic. It helps PfRv3 algorithm to choose the best channel to use for a given traffic class.

A domain level parameter is defined to store the probing information. You need to store two sets of parameters; general monitor and quick monitor. In other words, one can specify the number of packets to be sent in a probe burst and the interval between such bursts.

Smart probe are of three types:

- **Active Channel Probe**—Active channel probe is sent out to measure network delay if no probe is sent out for past 10 seconds interval.
- **Unreachable Probe**—Unreachable probe is used to detect channel reachability when there is no traffic send out.
- **Burst Probe**—Burst probes are used to calculate delay, loss, jitter on a channel that is not carrying active user traffic.

How to Configure PfRv3 Probe Reduction

Configuring PfRv3 Probe Reduction

You can perform this task on a hub master or a border device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **domain default**
4. Do one of the following:
   - **master hub**
   - **border**
5. **advanced**
6. **smart-probes burst** [**quick**] *number-of-packets* **packets every** *interval* **seconds**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**domain default** **Example:** `Device(config)# domain default`	Enters domain configuration mode.

Step 4	Do one of the following:
	• **master hub**
	• **border**
	Example:
	`Device(config-domain)# master hub`
	Example:
	`Device(config-domain)# border`
		Configures the device as a master hub and enters master controller configuration mode.
		Configures the device as the border and enters border configuration mode.
		Note If you select border configuration, it overwrites the master configuration.

Step 5	**advanced** **Example:** `Device(config-domain-mc)# advanced`	Enters advanced configuration mode.

	Command or Action	Purpose
	Example: `Device(config-domain-br)# advanced`	

Step 6	**smart-probes burst** [**quick**] *number-of-packets* **packets every** *interval* **seconds**
	Example:
	`Device(config-domain-mc-advanced)# smart-probe burst 10 packets every 20 seconds`
	Example:
	`Device(config-domain-br-advanced)# smart-probe burst quick 10 packets every 1 seconds`
		Specifies the number of packets to be sent in a probe burst and the interval between the bursts. The default values are as follows:
		• 1 packet every 1 second for default monitor
		• 20 packets every 1 second for quick monitor

Verifying PfRv3 Probe Reduction

SUMMARY STEPS

1. **show domain** {**default** | *domain-name*} [**vrf** *vrf-name*] {**master** | **border**} **status**

DETAILED STEPS

show domain {**default** | *domain-name*} [**vrf** *vrf-name*] {**master** | **border**} **status**

Use this command to verify the configuration.

Example:

```
Router# show domain default vrf green master status

Smart Probe Profile:
    General Monitor:
        Current Provision Level: Master Hub
            Master Hub:
                Packets per burst: 10
                Interval(secs): 20
    Quick Monitor:
        Current Provision Level: Master Hub
            Master Hub:
                Packets per burst: 10
                Interval(secs): 1
Smart Probe Inter-Packet Gap (ms) : 16
Smart Probe Timer Wheel Granularity (ms): 8
```

Configuration Examples for PfRv3 Probe Reduction

Example: PfRv3 Probe Reduction

```
domain default
 master hub
  advanced
   smart-probe burst 10 packets every 20 seconds
   smart-probe burst quick 10 packets every 1 seconds
```

Additional References for PfRv3 Probe Reduction

Related Documents

Related Topic	Document Title
Performance Routing Version 3 commands	Cisco IOS Performance Routing Version 3 Command Reference

CHAPTER 282

PfRv3 Intelligent Load Balance

The Performance Routing v3 (PfRv3) Intelligent Load Balance feature helps to move the traffic-classes based on the remote ingress interface, if the remote interface is overrun the bandwidth threshold. It validates remote interface ingress bandwidth when choosing the path. The PfRv3 Intelligent Load Balance feature detects the remote bandwidth overrun at the earliest and helps to reduce the packet drop caused by per tunnel QoS and increases the bandwidth utilization.

Feature Information for PfRv3 Intelligent Load Balance

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 338: Feature Information for PfRv3 Intelligent Load Balance

Feature Name	Releases	Feature Information
PfRv3 Intelligent Load Balance	Cisco IOS XE 16.11	The Performance Routing v3 (PfRv3) Intelligent Load Balance feature helps to move the traffic-classes (TC) based on the remote ingress interface, if the remote interface is overrun the bandwidth threshold. The following command was introduced: **remote-ingress-bandwidth-check**.

Prerequisites for PfRv3 Intelligent Load Balance

You must upgrade master hub software version to 16.11 or later. The spoke sites that require PfRv3 Intelligent Load Balance feature must be upgraded to version 16.11 or later. However, It is not mandatory to upgrade the spoke sites that do not use the PfRv3 intelligent Load Balance feature to the recommended versions.

Restrictions for PfRv3 Intelligent Load Balance

- The PfRv3 Intelligent Load balance supports the traffic only from hub to spoke.
- Only the default traffic classes are load-balanced among paths when the WAN interface is overrun in remote spoke sites.
- Remote bandwidth check is only supported on the hub or on the transit hub.
- Remote bandwidth TCA is sent from branch to hub or from branch to transit hub only.

Information About PfRv3 Intelligent Load Balance

How to Configure PfRv3 Intelligent Load Balance

Configuring PfRv3 Intelligent Load Balance

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **domain iwan**
4. **vrf default**
5. **master hub**
6. **load-balance**
7. **load-balance advanced**
8. **path-preference** *INET1* **fallback** *MPLS1*
9. **advanced**
10. **remote-ingress-bandwidth-check max** *75*
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**domain iwan** **Example:** `Device(config)# domain iwan`	Enters domain iwan configuration mode.
Step 4	**vrf default** **Example:** `Device(config-domain)# vrf default`	Configures Virtual Routing and Forwarding (VRF) for the default domain.
Step 5	**master hub** **Example:** `Device(config-domain-vrf)# master hub`	Configures the device as a master hub and enters master controller configuration mode.
Step 6	**load-balance** **Example:** `Device(config-domain-vrf-mc)# load-balance`	Enables load balance.
Step 7	**load-balance advanced** **Example:** `Device(config-domain-vrf-mc)# load balance advanced`	(Optional) Enables advanced mode of VRF on master hub.
Step 8	**path-preference** *INET1* **fallback** *MPLS1* **Example:** `Device(config-domain-vrf-mc-load-balance)# path-preference INET1 fallback MPLS1`	Specifies the path preference name and the fallback path(s) preference to use when the primary path(s) are out of policy.
Step 9	**advanced** **Example:** `Device(config-domain-vrf-mc-load-balance)# advanced`	Enters advanced configuration mode.
Step 10	**remote-ingress-bandwidth-check max** *75* **Example:** `Device(config-domain-vrf-mc-advanced)# remote-ingress-bandwidth-check max 75`	(Optional) Enables to change the value of remote bandwidth threshold. The default value of remote bandwidth threshold is 75%. You should change the remote bandwidth threshold followed by per tunnel QoS. **Note** Remote spoke site sends out BW-TCA, if WAN interface BW utilization exceeds the threshold.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Device(config-domain-vrf-mc-advanced)exit`	Exits border configuration mode and returns to privileged EXEC mode.

What to do next

The remote BW percentage must be configured after configuring PfRv3 Intelligent Load Balance.

Verifying PfRv3 Intelligent Load Balance

Use the following commands to verify PfRv3 intelligent load balance configuration:

- **show domain** *domain-name* **vrf** *vrf-name* **master exits**
- **show domain** *domain-name* **vrf** *vrf-name* **master exists** *site-id* **path-id**

Example: Configuring PfRv3 Intelligent Load Balance

```
domain iwan
 vrf default
 master hub
 load-balance advanced
 path-preference INET1 fallback MPLS1
 advanced
 remote-ingress-bandwidth-check max 75
```

Example: Verifying PfRv3 Intelligent Load Balance

The following is an example output from the **show domain iwan master exits** command.

```
  Device#show domain iwan master exits
 BR address: 168.254.0.2 | Name: Tunnel10 | type: external | Path: MPLS1 | path-id: 11 |
PLR TCs: 0


      Egress capacity: 1000000 Kbps | Egress BW: 2 Kbps | Ideal:1078 Kbps | under: 1076
Kbps | Egress Utilization: 0 %
      Ingress capacity: 1000000 Kbps | Ingress BW: 1076 Kbps | Ingress Utilization: 0 %

  BR address: 168.254.0.3 | Name: Tunnel20 | type: external | Path: INET1 | path-id: 12 |
PLR TCs: 0
      Egress capacity: 1000000 Kbps | Egress BW: 1076 Kbps | Ideal:1078 Kbps | under: 2
Kbps | Egress Utilization: 0 %
      Ingress capacity: 1000000 Kbps | Ingress BW: 2 Kbps | Ingress Utilization: 0 %
      DSCP: default[0]-Number of Traffic Classes[1]
```

The following is an example output from the **show domain iwan master exits 168.254.0.9 path-id** command.

```
  Device#domain iwan master exits 168.254.0.9 path-id
 Site id  : 168.254.0.9
Site mc type :  Branch
Border Address  : 168.254.0.9
     Service provider: MPLS1 path-id: 11 if_index: 28 bandwidth: 2000Kbps
bw-from-local-to-remote: 0Kbps  Address: NA
```

```
     Service provider: INET1 path-id: 12 if_index: 29 bandwidth: 300000Kbps
bw-from-local-to-remote: 1040Kbps  Address: NA
```

CHAPTER 283

Path Preference Hierarchy

The Path Preference Hierarchy feature allows you to configure service providers per VRF for traffic classes.

Feature Information for Path Preference Hierarchy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 339: Feature Information for Path Preference Hierarchy

Feature Name	Releases	Feature Information
Path Preference Hierarchy	Cisco IOS XE Denali 16.3.1	The following command was introduced or modified: **path-preference**.

Information About Path Preference Hierarchy

Overview of Path Preference Hierarchy

In an enterprise network, you would need to configure service providers to interconnect the hub and branches. The Path Preference Hierarchy feature allows you to configure three service providers per VRF for traffic classes. The service providers could be primary service provider, fallback service provider, and next-fallback service provider respectively. As the name suggests, the primary service provider is the first preference in the network, followed by fallback and next-fallback, respectively. You cannot have the same service provider for

primary and fallback as this results in a “fallback backhole.” In other words, each service provider must be unique.

Use the **path-preference** command to specify the service provider order. Use the **blackhole** or **routing** keywords for a next-fallback service provider to drop the packet if fallback is unavailable or to specify there is no next-fallback service provider, respectively. When a packet reaches “blackhole,” the packet is discarded.

How to Configure Path Preference Hierarchy

Configuring Path Preference Hierarchy

Perform this task to configure Path Preference Hierarchy feature on a hub.

```
domain default
 vrf green
  master hub
   source-interface Loopback1
   site-prefixes prefix-list HUBPFX
   class HEIRARCHICAL sequence 100
    match dscp ef policy custom
     priority 1 loss threshold 10
    path-preference ISP1 ISP2 fallback ISP3 next-fallback blackhole
```

The following is a sample output on a device that displays the route change reason and history. In this example, the traffic class jumps from next-fallback service provider to primary service provider, when the fallback is unavailable.

```
Dst-Site-Prefix: 100.30.0.0/16       DSCP: ef [46] Traffic class id:2
  Clock Time:                 12:57:15 (PST) 03/30/2015
  TC Learned:                 00:22:14 ago
  Present State:              CONTROLLED
  Current Performance Status: in-policy
  Current Service Provider:   ISP2 path-id:2 since 00:03:28
  Previous Service Provider:  ISP3 pfr-label: 0:0 | 0:7 [0x7] for 180 sec
  (A fallback/next-fallback provider. Primary provider will be re-evaluated 00:02:34 later)

  BW Used:                    3 Kbps
  Present WAN interface:      Tunnel20 in Border 100.10.2.1
  Present Channel (primary):  46 ISP2 pfr-label:0:0 | 0:2 [0x2]
  Backup Channel:             42 ISP3 pfr-label:0:0 | 0:7 [0x7]
  Destination Site ID bitmap: 0
  Destination Site ID:        100.30.1.1
  Class-Sequence in use:      10
  Class Name:                 BUSINESS using policy User-defined
    priority 2 packet-loss-rate threshold 10.0 percent
    priority 2 byte-loss-rate threshold 10.0 percent
  BW Updated:                 00:00:14 ago
  Reason for Latest Route Change:    next-fallback to Higher Path Preference
  Route Change History:
              Date and Time                    Previous Exit                          Current
 Exit                      Reason

    1:  12:53:47 (PST) 03/30/2015   ISP3/100.10.1.1/Tu30 (Ch:42)
ISP2/100.10.2.1/Tu20 (Ch:46)            next-fallback to Higher Path Preference
    2:  12:50:47 (PST) 03/30/2015   None/0.0.0.0/None (Ch:0)
ISP3/100.10.1.1/Tu30 (Ch:42)            Uncontrolled to Controlled Transition
    3:  12:50:15 (PST) 03/30/2015   ISP3/100.10.1.1/Tu30 (Ch:42)          None/0.0.0.0/None
 (Ch:0)                No Channels Available
    4:  12:48:14 (PST) 03/30/2015   ISP2/100.10.4.1/Tu20 (Ch:43)
```

```
ISP3/100.10.1.1/Tu30 (Ch:42)            Exit down
    5:  12:47:57 (PST) 03/30/2015   ISP2/100.10.2.1/Tu20 (Ch:46)
ISP2/100.10.4.1/Tu20 (Ch:43)            Exit down
```

In the following example, continuation of the above example, the traffic class is now controlled by primary service provider.

```
 Route Change History:
             Date and Time                  Previous Exit                          Current
 Exit                      Reason

    1:  12:59:49 (PST) 03/30/2015   ISP2/100.10.2.1/Tu20 (Ch:46)
ISP1/100.10.1.1/Tu10 (Ch:41)            Backup to Primary path preference transition
    2:  12:53:47 (PST) 03/30/2015   ISP3/100.10.1.1/Tu30 (Ch:42)
ISP2/100.10.2.1/Tu20 (Ch:46)            next-fallback to Higher Path Preference
    3:  12:50:47 (PST) 03/30/2015   None/0.0.0.0/None (Ch:0)
ISP3/100.10.1.1/Tu30 (Ch:42)            Uncontrolled to Controlled Transition
    4:  12:50:15 (PST) 03/30/2015   ISP3/100.10.1.1/Tu30 (Ch:42)          None/0.0.0.0/None
 (Ch:0)                No Channels Available
    5:  12:48:14 (PST) 03/30/2015   ISP2/100.10.4.1/Tu20 (Ch:43)
ISP3/100.10.1.1/Tu30 (Ch:42)            Exit down
```

In the following example, continuation of the above example, the traffic class is discarded since the packet has reached a blackhole.

```
Route Change History:
             Date and Time                  Previous Exit                          Current
 Exit                      Reason

   1:  12:50:15 (PST) 03/30/2015   ISP3/100.10.1.1/Tu30 (Ch:42)          None/0.0.0.0/None
 (Ch:0)                No Channels Available
    2:  12:48:14 (PST) 03/30/2015   ISP2/100.10.4.1/Tu20 (Ch:43)
ISP3/100.10.1.1/Tu30 (Ch:42)            Exit down
    3:  12:47:57 (PST) 03/30/2015   ISP2/100.10.2.1/Tu20 (Ch:46)
ISP2/100.10.4.1/Tu20 (Ch:43)            Exit down
    4:  12:44:42 (PST) 03/30/2015   ISP1/100.10.1.1/Tu10 (Ch:41)
ISP2/100.10.2.1/Tu20 (Ch:46)            Exit down
    5:  12:44:13 (PST) 03/30/2015   ISP1/100.10.3.1/Tu10 (Ch:44)
ISP1/100.10.1.1/Tu10 (Ch:41)            Exit down
```

Additional References for Path Preference Hierarchy

Related Documents

Related Topic	Document Title
Performance Routing Version 3 commands	Cisco IOS Performance Routing Version 3 Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Path Preference Hierarchy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 340: Feature Information for Path Preference Hierarchy

Feature Name	Releases	Feature Information
Path Preference Hierarchy	Cisco IOS XE Denali 16.3.1	The following command was introduced or modified: **path-preference**.

CHAPTER 284

PfRv3 Remote Prefix Tracking

Performance Routing Version 3 (PfRv3) is an intelligent-path control mechanism for improving application delivery and WAN efficiency. The PfRv3 Remote Prefix Tracking feature enhances networks running Performance Routing Version 3 (PfRv3) to learn the prefix of a remote device from the Routing Information Base (RIB) table.

Feature Information for PfRv3 Remote Prefix Tracking

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 341: Feature Information for PfRv3 Remote Prefix Tracking

Feature Name	Releases	Feature Information
PfRv3 Remote Prefix Tracking	Cisco IOS release 3.16.6, 15.6M2, 15.5.3M6, 15.7M, 16.3.5, and Cisco IOS XE Everest 16.6.1.	The following command was modified: **show domain default vrf**.

Information About PfRv3 Remote Prefix Tracking

Site Prefixes Database

Site Prefixes are LAN side prefixes owned by each site. The site prefix database is central to the site concept in PfRv3. Site prefix database reside on the master controller.

- The master site learns the remote site prefix through SAF advertised by remote MC. Master site learns the local site prefix from the local borders. The border learns the prefix from RIB and sends the prefix learned to the local master
- The border site prefix database is populated by SAF messages published by all the remote site master and local site master.
- By default, MCs and BRs delete site prefixes every 24 hours.

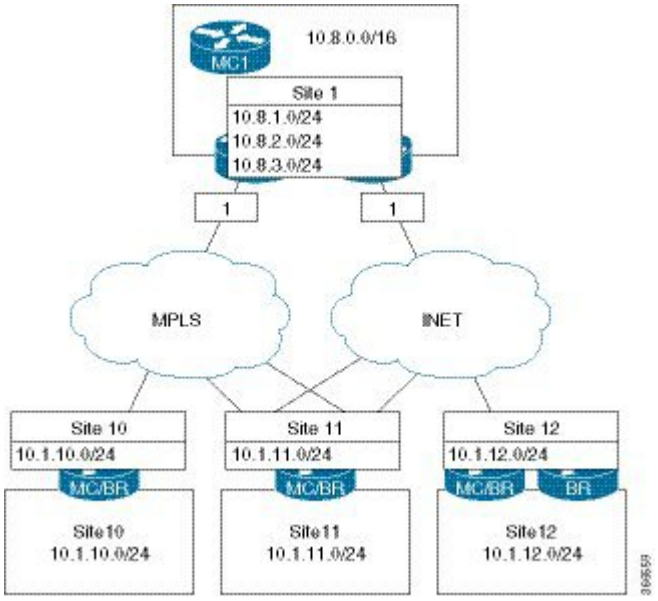
Learning Local Site Prefixes

Border routers collect the prefix from the RIB table and send it to the local master controller. After receiving prefixes from a border router, the local master controller filters prefixes as per the following criteria.

1. If a prefix is learned on a tunnel interface, the prefix is marked remote and not added to local LAN list.
2. If a prefix is learned from NHRP, the prefix is not added to LAN list.
3. If a prefix is learned on a physical interface of the tunnel interface. the prefix is not added to LAN list.
4. If an enterprise prefix is configured on the hub and the prefix is part of the enterprise prefix list configured on hub, the branch master adds the prefix from the RIB table to the LAN list.

The prefixes in the LAN list are added to the site prefix database as local site prefix list.

Figure 236: Learning Local Site Prefixex

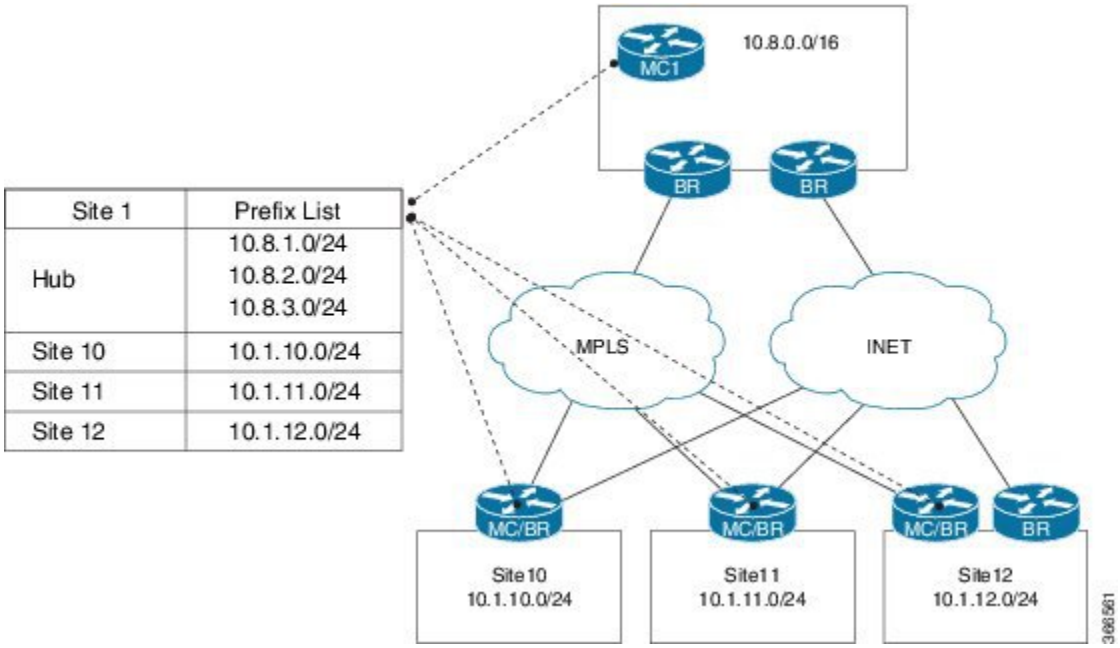


Learning Remote Site Prefixes

In order to learn from advertisements via the peering infrastructure from remote peers, every MC and BR subscribes to the peering service for the subservice of site prefix. MCs publish and receive site prefixes. BRs only receive site prefixes. MC learns prefixes from the border and filters the prefixes as explained in the previous section and publishes the prefixex to all sites. This message is received by all MCs and BRs that

subscribe to the peering service. The message is decoded and added to the site prefix databases at those MCs and BRs.

Figure 237: Pfrv3-discovery-site-prefix.png



PfRv3 Remote Prefix Tracking via Egress Flow

Prior to Cisco IOS XE Everest 16.6.1, the site prefix was learnt via the egress flow on the WAN interface. The prefix thus, learnt is published to all remote sites in the network using the EIGRP SAF message. If a remote site does not receive a new SAF message within 24 hours, the prefix is removed from the local-prefix database. If the routing is updated within 24 hours, corresponding prefix table will not be updated. Since, the prefix is learned from the egress traffic, sometimes-wrong prefixes are learnt due to redirected traffic. These wrongly learnt prefixes are not cleaned up until the 24 hour age out time.

Additionally, the prefix reachability is not tracked per channel. For example, if the prefix belongs to a specific site, it is assumed that prefix is reachable through all the channels available for that site. This results in a traffic blackhole when the prefix is not reachable through the selected channel.

PfRv3 Remote Prefix Tracking via RIB table

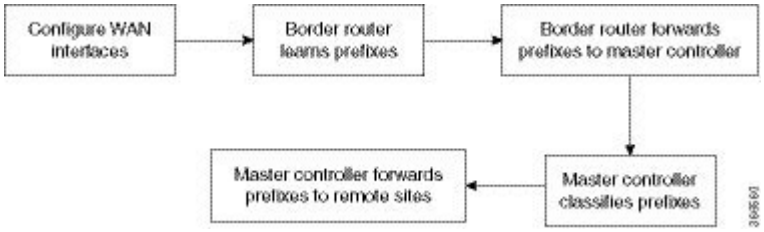
The PfRv3 Remote Prefix Tracking feature prevents the above scenarios by learning the local site prefixes from the RIB table instead of the egress flow. The prefixes are advertised to the remote sites. Changes to RIB table are tracked and are accordingly notified to all remote sites. Therefore, all sites are updated automatically with the precise site prefix information. Remote site tracks the prefix learnt via the WAN interface. While controlling the traffic, remote sites validate the reachability of the prefix on all channels available for a site.

There is no specific configuration required for this feature. You only need to configure the WAN interfaces.

How Site Prefix is Learnt?

The following workflow illustrates the process of how site prefix is learnt.

Figure 238: Site Prefix Learning Workflow



WAN Interfaces Configuration

You must configure the WAN interfaces on a border router in a branch using the **domain***domain-name* **dynamic-path** command. For more information, see "Configuring Branch Border Router" in the *Performance Routing Version 3* chapter.

Prefix Learning on Border Router

On initialization, the border device learns the entire prefix from the RIB table and stores in the local prefix database, where the information is classified per VRF. Any changes in the RIB database, such as addition or deletion of prefixes, are accounted in the prefix database as appropriate. Prefixes learned from the RIB on the local border are forwarded to the local master controller. The prefix information in the border device can be viewed using the **show domain default vrf** *vrf name* **border route-import** command.

Forwarding the Prefix to Master Controller

Master controller learns about a new prefix added or removed in the RIB table from the border device.

On a branch site, when the WAN interfaces are configured using the **domain***domain-name* **dynamic-path** command., the wan interface details are shared with the master controller by all border routers in a site. The master controller classifies this prefix information as WAN or LAN prefix, as appropriate.

On a hub site, The prefixes are learnt and classified similar to a branch site. The only difference is the command used to configure the WAN interface, which is **domain path service-provider-name path-id** *number* command.

Note It is mandatory to configure prefixes on the hub and the transit hub. It is also mandatory to configure the **domain** *domain-name* **dynamic-path** in branch tunnel interface.

Prefix Classification by Master Controller

Master controller filters the prefix using the criteria described in the *Learning Local Site Prefixes* section and updates the local prefix database. The local prefix database is published to all the subscribers using the EIGRP SAF message.The prefix information in the border device can be viewed using the following commands:

- **show domain** {*domain-name* | **default**} **vrf** *vrf-name* **master route-import local all**
- **show domain** {*domain-name* | **default**} **vrf** *vrf-name* **master route-import border** *border-ip*

- **show domain** {*domain-name* | **default**} **vrf** *vrf-name* **master route-import local**
- **show domain** {*domain-name* | **default**} **vrf** *vrf-name* **master route-import remote**
- **show domain** {*domain-name* | *default*} **vrf** *vrf-name* **border route-import**
- **show domain** {*domain-name* | *default*} **vrf** *vrf-name* **border local-prefix interface** *interface-name*

Path Preference

When a master controller receives prefixes from a border router, the master controller evaluates the traffic classes to a device, whose prefixes are listed in the RIB table and performs a policy decision to select a channel.

A channel is added to a channel list of a traffic class when a device associated with a prefix is reachable. The master controller decides on a path to a device based on the reachability of device (with a prefix in the RIB) on a channel. Prefixes are validated as follows:

- The list of interfaces on which prefixes are reachable is obtained from the prefix database and the prefix is verified for reachability via the same interface as the channel interface.
- A list of routes is obtained for a prefix that is reachable via an interface.

The channel is verified for the next hop address and if the next hop matches the appropriate prefix route. If the parent route of a device pertaining to a prefix matches the channel next hop, it indicates that the device with the prefix is reachable through a channel. If prefixes cannot be reached on a channel, a syslog message is displayed.

Note Maximum secondary paths must be configured on the border devices using the maximum-paths command so that prefixes are reachable. This command are enabled in the EIGRP or BGP router configuration mode.

How to Display Site Prefixes

Displaying Site Prefixes Learnt By a Border Router

SUMMARY STEPS

1. **show domain** *domain-name* **vrf** *vrf-name* **border site-prefix**
2. **show domain default vrf** *vrf name* **border route-import**
3. **show domain default vrf** *vrf name* **border route-import interface**
4. **show monitor event-trace pfrv3 all**

DETAILED STEPS

Step 1 **show domain** *domain-name* **vrf** *vrf-name* **border site-prefix**

Use this command to verify the reachability of the prefix on all channels.

Step 2 **show domain default vrf** *vrf name* **border route-import**

Use this command to view the prefix information learnt by a border device from the RIB table.

Example:

```
B1MCBR# show domain default vrf green border route-import

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Proto Prefix              Location   Next-Hop          Index      Interface                 In-RIB
-------------------------------------------------------------------------------------------------
L     10.20.0.1/32        Local      0.0.0.0           29         Ethernet0/2.30             YES
C     10.20.0.0/24        Local      0.0.0.0           29         Ethernet0/2.30             YES
L     10.20.1.1/32        Local      0.0.0.0           25         Ethernet0/1.30             YES
C     10.20.1.0/24        Local      0.0.0.0           25         Ethernet0/1.30             YES
D     10.20.2.0/24        Local      10.20.0.2         29         Ethernet0/2.30             YES
L     51.1.0.4/32         Local      0.0.0.0           24         Tunnel10                   YES
C     51.1.0.0/16         Local      0.0.0.0           24         Tunnel10                   YES
D     52.1.0.0/16         Local      10.20.0.2         29         Ethernet0/2.30             YES
C     100.20.1.1/32       Local      0.0.0.0           22         Loopback1                  YES
D     100.20.2.1/32       Local      10.20.0.2         29         Ethernet0/2.30             YES
S     100.20.3.1/32       Local      10.20.0.3         29         Ethernet0/2.30             YES
```

Step 3 **show domain default vrf** *vrf name* **border route-import interface**

Use this command to view the prefix information associated with an interface.

Example:

```
B1MCBR# show domain default vrf green border route-import interface Loopback1

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Proto Prefix              Location   Next-Hop          Index      Interface                 In-RIB
-------------------------------------------------------------------------------------------------
C     100.20.1.1/32       Local      0.0.0.0           22         Loopback1                  YES
```

Step 4 **show monitor event-trace pfrv3 all**

Enables debugging by collecting trace.

Displaying Site Prefixes Learnt By a Master Controller

SUMMARY STEPS

1. **show domain default vrf** *vrf name* **master route-import**
2. **show domain default vrf** *vrf name* **master route-import interface**
3. **show domain default vrf** *vrf name* **master local-prefix**

DETAILED STEPS

Step 1 show domain default vrf *vrf name* master route-import

Use this command to view the prefix information learnt by a master controller.

Example:

```
B1MCBR# show domain default vrf green master route-import all

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
------------------------------------------------------------------------------------------------------------------------
Enterprise Prefix List:
Prefix: 100.20.0.0, Mask: 16
Prefix: 100.30.0.0, Mask: 16
Prefix: 100.0.0.0, Mask: 8
------------------------------------------------------------------------------------------------------------------------
Proto Prefix               Location   BR-IP             Next-Hop          Index      Interface
             IF-Role   In-RIB
------------------------------------------------------------------------------------------------------------------------
B     10.10.1.0/24         Remote     100.20.1.1        10.20.1.2         25         Ethernet0/1.30
             WAN       YES
B     10.10.2.0/24         Local      100.20.2.1        10.20.2.2         25         Ethernet0/1.32
             LAN       YES
B     10.10.3.0/24         Remote     100.20.1.1        10.20.1.2         25         Ethernet0/1.30
             WAN       YES
B     10.10.4.0/24         Local      100.20.2.1        10.20.2.2         25         Ethernet0/1.32
             LAN       YES
B     10.15.1.0/24         Remote     100.20.1.1        10.20.1.2         25         Ethernet0/1.30
             WAN       YES
B     10.15.2.0/24         Local      100.20.2.1        10.20.2.2         25         Ethernet0/1.32
             LAN       YES
L     10.20.0.1/32         Local      100.20.1.1        0.0.0.0           29         Ethernet0/2.30
             LAN       YES
L     10.20.0.2/32         Local      100.20.2.1        0.0.0.0           28         Ethernet0/2.30
             LAN       YES
C     10.20.0.0/24         Local      100.20.1.1        0.0.0.0           29         Ethernet0/2.30
             LAN       YES
C     10.20.0.0/24         Local      100.20.2.1        0.0.0.0           28         Ethernet0/2.30
             LAN       YES
L     10.20.1.1/32         Remote     100.20.1.1        0.0.0.0           25         Ethernet0/1.30
             WAN       YES
C     10.20.1.0/24         Remote     100.20.1.1        0.0.0.0           25         Ethernet0/1.30
             WAN       YES
```

```
D     10.20.1.0/24          Remote     100.20.2.1        10.20.0.1         28         Ethernet0/2.30
            LAN       YES
L     10.20.2.1/32          Local      100.20.2.1        0.0.0.0           25         Ethernet0/1.32
            LAN       YES
D     10.20.2.0/24          Local      100.20.1.1        10.20.0.2         29         Ethernet0/2.30
            LAN       YES
C     10.20.2.0/24          Local      100.20.2.1        0.0.0.0           25         Ethernet0/1.32
            LAN       YES
B     10.30.1.0/24          Remote     100.20.1.1        10.20.1.2         25         Ethernet0/1.30
            WAN       YES
B     10.30.2.0/24          Local      100.20.2.1        10.20.2.2         25         Ethernet0/1.32
            LAN       YES
L     51.1.0.4/32           Remote     100.20.1.1        0.0.0.0           24         Tunnel10
            WAN       YES
C     51.1.0.0/16           Remote     100.20.1.1        0.0.0.0           24         Tunnel10
            WAN       YES
```

```
B1MCBR# show domain default vrf green master route-import local

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
------------------------------------------------------------------------------------------------------------------------
Enterprise Prefix List:
Prefix: 100.20.0.0, Mask: 16
Prefix: 100.30.0.0, Mask: 16
Prefix: 100.0.0.0, Mask: 8
------------------------------------------------------------------------------------------------------------------------
Proto Prefix              Location   BR-IP             Next-Hop          Index      Interface
            IF-Role   In-RIB
------------------------------------------------------------------------------------------------------------------------
L     10.20.0.1/32          Local      100.20.1.1        0.0.0.0           29         Ethernet0/2.30
            LAN       YES
C     10.20.0.0/24          Local      100.20.1.1        0.0.0.0           29         Ethernet0/2.30
            LAN       YES
C     10.20.0.0/24          Local      100.20.2.1        0.0.0.0           28         Ethernet0/2.30
            LAN       YES
D     10.20.2.0/24          Local      100.20.1.1        10.20.0.2         29         Ethernet0/2.30
            LAN       YES
C     10.20.2.0/24          Local      100.20.2.1        0.0.0.0           25         Ethernet0/1.32
            LAN       YES
C     100.20.1.1/32         Local      100.20.1.1        0.0.0.0           22         Loopback1
            LAN       YES
D     100.20.1.1/32         Local      100.20.2.1        10.20.0.1         28         Ethernet0/2.30
            LAN       YES
D     100.20.2.1/32         Local      100.20.1.1        10.20.0.2         29         Ethernet0/2.30
            LAN       YES
C     100.20.2.1/32         Local      100.20.2.1        0.0.0.0           23         Loopback1
            LAN       YES
S     100.20.3.1/32         Local      100.20.1.1        10.20.0.3         29         Ethernet0/2.30
            LAN       YES
```

```
B1MCBR# show domain default vrf green master route-import remote

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
```

```
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
------------------------------------------------------------------------------------------------------------------------
Enterprise Prefix List:
Prefix: 100.20.0.0, Mask: 16
Prefix: 100.30.0.0, Mask: 16
Prefix: 100.0.0.0, Mask: 8
------------------------------------------------------------------------------------------------------------------------
Proto Prefix              Location   BR-IP            Next-Hop          Index      Interface
             IF-Role   In-RIB
------------------------------------------------------------------------------------------------------------------------
L     10.20.1.1/32        Remote     100.20.1.1       0.0.0.0           25         Ethernet0/1.30
             WAN       YES
C     10.20.1.0/24        Remote     100.20.1.1       0.0.0.0           25         Ethernet0/1.30
             WAN       YES
L     51.1.0.4/32         Remote     100.20.1.1       0.0.0.0           24         Tunnel10
             WAN       YES
C     51.1.0.0/16         Remote     100.20.1.1       0.0.0.0           24         Tunnel10
             WAN       YES
D     52.1.0.0/16         Remote     100.20.1.1       10.20.0.2         29         Ethernet0/2.30
             LAN       NO
D     52.1.0.0/16         Remote     100.20.1.1       51.1.0.3          24         Tunnel10
             WAN       YES
B     10.10.1.0/24        Remote     100.20.1.1       10.20.1.2         25         Ethernet0/1.30
             WAN       YES
B     10.10.3.0/24        Remote     100.20.1.1       10.20.1.2         25         Ethernet0/1.30
             WAN       YES
B     10.15.1.0/24        Remote     100.20.1.1       10.20.1.2         25         Ethernet0/1.30
             WAN       YES
B     10.30.1.0/24        Remote     100.20.1.1       10.20.1.2         25         Ethernet0/1.30
             WAN       YES
D     100.10.0.0/16       Remote     100.20.1.1       10.20.0.2         29         Ethernet0/2.30
             LAN       NO
D     100.10.0.0/16       Remote     100.20.1.1       51.1.0.2          24         Tunnel10
             WAN       YES

B1MCBR# show domain default vrf green master route-import border 100.20.1.1

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
------------------------------------------------------------------------------------------------------------------------
Enterprise Prefix List:
Prefix: 100.20.0.0, Mask: 16
Prefix: 100.30.0.0, Mask: 16
Prefix: 100.0.0.0, Mask: 8
------------------------------------------------------------------------------------------------------------------------
Proto Prefix              Location   BR-IP            Next-Hop          Index      Interface
             IF-Role   In-RIB
------------------------------------------------------------------------------------------------------------------------
L     10.20.0.1/32        Local      100.20.1.1       0.0.0.0           29         Ethernet0/2.30
             LAN       YES
C     10.20.0.0/24        Local      100.20.1.1       0.0.0.0           29         Ethernet0/2.30
             LAN       YES
L     10.20.1.1/32        Remote     100.20.1.1       0.0.0.0           25         Ethernet0/1.30
             WAN       YES
C     10.20.1.0/24        Remote     100.20.1.1       0.0.0.0           25         Ethernet0/1.30
```

```
              WAN       YES
D     10.20.2.0/24          Local      100.20.1.1        10.20.0.2         29          Ethernet0/2.30
              LAN       YES
L     51.1.0.4/32           Remote     100.20.1.1        0.0.0.0           24          Tunnel10
              WAN       YES
C     51.1.0.0/16           Remote     100.20.1.1        0.0.0.0           24          Tunnel10
              WAN       YES
D     52.1.0.0/16           Remote     100.20.1.1        10.20.0.2         29          Ethernet0/2.30
              LAN       NO
D     52.1.0.0/16           Remote     100.20.1.1        51.1.0.3          24          Tunnel10
              WAN       YES
C     100.20.1.1/32         Local      100.20.1.1        0.0.0.0           22          Loopback1
              LAN       YES
D     100.20.2.1/32         Local      100.20.1.1        10.20.0.2         29          Ethernet0/2.30
              LAN       YES
```

Step 2 **show domain default vrf** *vrf name* **master route-import interface**

Use this command to view the prefix information associated with an interface.

Example:

```
Router# show domain default vrf green border local-prefix interface Ethernet0/0.10
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
       B-BGP D - EIGRP, EX - EIGRP external, O - OSPF,
       IA - OSPF inter area N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type 2 E1 - OSPF external type 1,
       E2 -   OSPF external type 2 i - IS-IS, su - IS-IS summary,
       L1 - IS-IS level-1, L2 - IS-IS level-2 ia - IS-IS inter area,
       * - candidate default, H- NHRP
Local  -- Prefix learned over LAN.
Remote -  Prefix learned over WAN.
Prefix          Interface       BR IP       Index   Prefix-site  Proto Next-Hop        Status
-------------------------------------------------------------------------
100.10.4.1/32   Ethernet0/0.10  100.20.1.1  12      Local         C                     -----
    Up
```

Step 3 **show domain default vrf** *vrf name* **master local-prefix**

Use this command to view the prefix information associated with an border router.

Example:

```
Router# show domain default vrf green master local-prefix border-ip 100.20.1.1
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
       B-BGP D - EIGRP, EX - EIGRP external, O - OSPF,
       IA - OSPF inter area N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type 2 E1 - OSPF external type 1,
       E2 -   OSPF external type 2 i - IS-IS, su - IS-IS summary,
       L1 - IS-IS level-1, L2 - IS-IS level-2 ia - IS-IS inter area,
       * - candidate default, H- NHRP
Local  -- Prefix learned over LAN.
Remote -  Prefix learned over WAN.
Prefix        Interface        BR IP       Index   Prefix-site  Proto      Next-Hop     Status
-------------------------------------------------------------------------
100.10.4.1/32  Ethernet0/0.10  100.20.1.1  12      Local        C
```

Additional References for PfRv3 Remote Prefix Tracking

Related Documents

Related Topic	Document Title
PfRv3commands	Cisco IOS Performance Routing Version 3 Command Reference
Site Prefix Splitting	Site Prefix Splitting

CHAPTER 285

PfRv3 Per Interface Probe Tuning

The PfRv3 Per Interface Probe Tuning feature provides the flexibility to specify different profiles for a channel associated with an interface thereby allowing you to measure the metrics of a channel.

Feature Information for PfRv3 Per Interface Probe Tuning

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 342: Feature Information for PfRv3 Per Interface Probe Tuning

Feature Name	Releases	Feature Information
PfRv3 Per Interface Probe Tuning	Cisco IOS XE Everest 16.6.1	The following commands were introduced or modified: **domain smart-probe**, **smart-probe**, **show platform hardware qfp active feature pfrv3**, **show platform software pfrv3**.

Prerequisites for PfRv3 Probe Reduction

Restrictions for PfRv3 Per Interface Probe Tuning

- The profile parameters must be defined or enforced on all border hub routers. Configuring the profile on a hub master controller does not propagate the profile parameters to the border hub routers.
- The default data expiration value for a channel is 90 seconds.
- You must configure the Performance Routing v3 Zero SLA Support feature on the hub border router to suppress nonzero DSCP (Differentiated Services Code Point) channels.

Information About PfRv3 Per Interface Probe Tuning

Probe Reduction and Per Interface Probe Tuning

Probing helps in measuring the metrics of a channel. A “profile” is a set of probing parameters configured on a device to send a probe packet on a channel that must be monitored. Before sending a probe packets on a channel, the channel that is to be monitored must be understood because each monitor has different profiles. In most cases, there are two monitors—default and quick. Each probe has two parameters, namely, burst packets and burst interval, which can be configured to define the probe packets sent on a PfR channel.

The PfRv3 Probe Reduction feature allows reducing traffic probe on channels that do not carry any traffic. For more information see the *PfRv3 Probe Reduction* module.

The PfRv3 Probe Reduction feature enforces similar probing on all interfaces irrespective of an interface through which a channel goes out, whereas the PfRv3 Per Interface Probe Tuning feature provides the flexibility to enforce different profiles on channels associated with an interface.

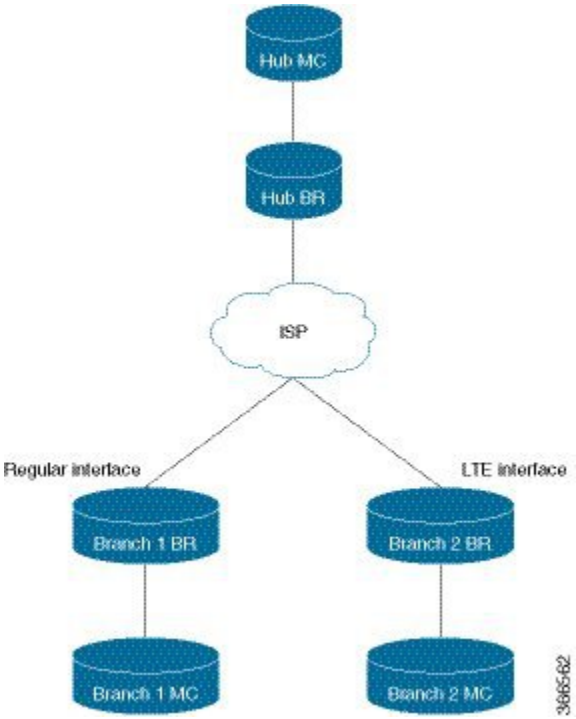
How Per Interface Probe Tuning Works?

The PfRv3 Per Interface Probe Tuning feature is configured on border hub routers via the *profile-id* argument in the **smart-probes** command and applied to an interface via the **domain smart-probe profile** command.

If you do not configure these commands, the default profile 0, is set on a device. The default profile has predefined parameters of 1 packet every 1 second for a default monitor and 20 packets every 1 second for a quick monitor.

The following is a sample topology to explain the working of the PfRv3 Per Interface Probe Tuning feature.

Figure 239: Per Interface Probe Tuning—Sample Topology



A hub branch router communicates to two branch routers Branch 1 Router and Branch 2 Router via ISP. Branch 1 Router has a regular interface, while Branch 2 Router has an Long-Term Evolution (LTE) interface. The LTE interface requires different probing parameters on the channel connected to the interface as LTE radio channels are established when data needs to be transmitted over the interface and radio frequency band occupies the transmission.

The profile parameters for the LTE interface are 100 packets every 1200 seconds for default monitor and default values for quick monitor. The profile parameters for the regular interface is the default parameters, which is, one packet every one second for default monitor and 20 packets every one second for quick monitor.

The hub border router establishes channels through its WAN interface to Branch 1 Router and Branch 2 Router via the ISP. Based on the defined profile parameters, channels from the hub border router to the Branch 1 Router are probed at regular intervals. Channels from the hub border router to Branch 2 Router wil have not have incoming probes for 19 minutes. The following happens before data is transmitted to the LTE interface:

- Burst probe packets are sent over the channel to measure the metrics.
- The burst interval range is increased to allow a longer duration so that radio bandwidth is not stagnated.
- Unreachable probe packets are not sent after sending the burst probe packets. This is to free up the radio bandwidth. to transmit the data.
- The burst interval range is configured to a longer duration so that radio bandwidth is not occupied.
- Unreachability detection is suppressed to ensure that there is no unreachability from a remote device for a period of time.

Profile—Channel Association

The profiles are associated with the channel and not the interface because it is possible that the same interface may host different channels, especially on border hub routers. If two channels have different profile numbers, the channel with a higher profile number is chosen to transmit data. The profile negotiation rule requires a profile with higher ID number to have a slower probing rate. The default profile (one packet every one second for default monitor and 20 packets every one second for quick monitor) has sufficient probing rate.When a channel probes at a slower rate (bigger profile ID number) another channel in the network probes at a higher rate (smaller profile ID number).

Note There is no automatic detection mechanism to calculate the rate of different profiles if the profile negotiation rule (higher-ID-slower-rate) is violated.

How to Configure PfRv3 Per Interface Probe Tuning

Defining a Profile on a Border Hub Router

```
domain domain1
 border
  advanced
   smart-probe 1 burst quick 10 packets every 20 seconds 1
```

Applying a Profile to an Interface on a Border Hub Router

```
interface tunnel 100
 domain smart-probe profile 1
```

Verifying Profile Parameters

The following is a sample output of the **show platform software pfrv3** command that displays the profile parameters applied to an device:

```
HubBr2# show platform software pfrv3 rp active smart-probe
PfRv3 smart probe parameters :
Profile ID: 0
Attribute: 0x0000
Probe Burst interval: 1 second
Probe Burst number: 1 packets
Quick Monitor Probe Burst interval: 1 second
Quick Monitor Probe Burst number: 20 packets
Unreachable interval: 4 second
Profile ID: 1
Attribute: 0x0000
Probe Burst interval: 0 second
Probe Burst number: 0 packets
Quick Monitor Probe Burst interval: 0 second
Quick Monitor Probe Burst number: 0 packets
Unreachable interval: 4 second
Profile ID: 2
Attribute: 0x0000
```

```
Probe Burst interval: 0 second
Probe Burst number: 0 packets
Quick Monitor Probe Burst interval: 0 second
Quick Monitor Probe Burst number: 0 packets
Unreachable interval: 4 second
```

Verifying Profile Parameters Associated with a Channel

The following is a sample output of the **show platform hardware qfp** command that displays the profile parameters associated with a channel:

```
Branch100# show platform hardware qfp active feature pfrv3 client channel id 7 detail
Chan id: 7 tbl-id: 0, if_h: 14(Tunnel100), site-id: 10.3.1.1, in_uidb: 65528, dscp: 0,
pfr-label: 0:0 | 0:0 [00000000]
  Supports zero-sla: Yes
  Muted by zero-sla: No
  Plr rx state: No
  Plr tx state: No
  Plr establish state: No
  next hop: 100.1.1.1
  State:    Discovered and open
  rx state: Reachable
  tx state: Reachable
  Smart Probe in Burst: No
  Unreach Probing only: Off
  Profile_ID: 0
  V4 Smart Probe Received: Yes
  V4 Smart Probe Sent: Yes
  Current profile_id: 1 <<< different than "Profile ID" (two lines above), resulted from
negotiation
  Remote profile_id: 1
  hash val: 25699
  exmem info:
        PPE addr: 0xebd26000
  stats:
    RX pkts: 0  bytes: 0
    TX pkts: 0  bytes: 0
    Blackhole pkts: 0  bytes: 0
    Loop pkts: 0  bytes: 0
    Probes: rx: 6288 tx: 474
    Number of SMP Profile Bursts sent: 100
    Number of Active Channel Probes sent: 374
    Number of Reachability Probes sent: 0
    Number of Force Unreaches sent: 0
    Last Probe rx: 44115 ms Ago
    Last Probe tx: 3379 ms Ago
```

Configuration Examples for PfRv3 Per Interface Probe Tuning

Additional References for PfRv3 Per Interface Probe Tuning

Related Documents

Related Topic	Document Title
Performance Routing Version 3 commands	Cisco IOS Performance Routing Version 3 Command Reference
Probe Reduction	PfRv3 Probe Reduction

Standards and RFCs

Standard/RFC	Title

CHAPTER 286

PfRv3 Inter-DC Optimization

The PfRv3 Inter-DC (IDC) Optimization feature optimizes traffic between hub and transit hub sites over a WAN overlay or a DCI overlay. A path-preference policy specific to inter-DC Optimization is used for optimizing traffic between two or more hub sites. The PfRv3-Inter-DC-Optimization routs traffic from a hub site to another hub site for specific traffic types such as data, voice, video, and so on.

Feature Information for PfRv3 Inter-DC Optimization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 343: Feature Information for PfRv3 Inter-DC Optimization

Feature Name	Releases	Feature Information
PfRv3 Inter-DC Optimization	Cisco IOS XE Everest 16.6.1	The following commands were introduced or modified: **domain**, **inter-dc**, **interdc-path-preference**.

Prerequisites for PfRv3 Inter-DC Optimization

- Hub sites must be upgraded for using the same version of IOS for the master and border devices.
- Static NHRP mapping must be used between hub sites. (NHRP shortcuts are not allowed between hub sites)

- Local LAN prefixes on each hub site (all borders) must have a specific route pointing to LAN interfaces and not to DCI or WAN interfaces.

Limitations and Guidelines for Inter-DC Optimization

- The PfRv3 Inter-DC Optimization does not optimize routes using common prefixes.

Note A common prefix is a prefix which is configurared as a static prefix on all the hub sites, that include hub sites and transite hub sites.

- The command **domain dci-path** should be added in DCI tunnel interface, but normal WAN interface with **domain path** command can also be chosen as DCI path. But DCI interface using **domain dci-path** cannot be chosen as the path for normal hub to spoke traffic.
- We recommend to use static configuration under DCI tunnel interface to set up peer between DC sites. If **nhrp shortcut** is used, a forwarding loop may occur.
- After enabling the IDC feature using the inter-dc command, you can configure **path-preference** and **interdc-path-preference** under policy.

Note You should not configure DCI path in **global path-preference** because if you add DCI path into path-preference, there is no channel available between hub and spoke in the DCI path. The DCI path cannot be chosen for the normal traffic-classes.

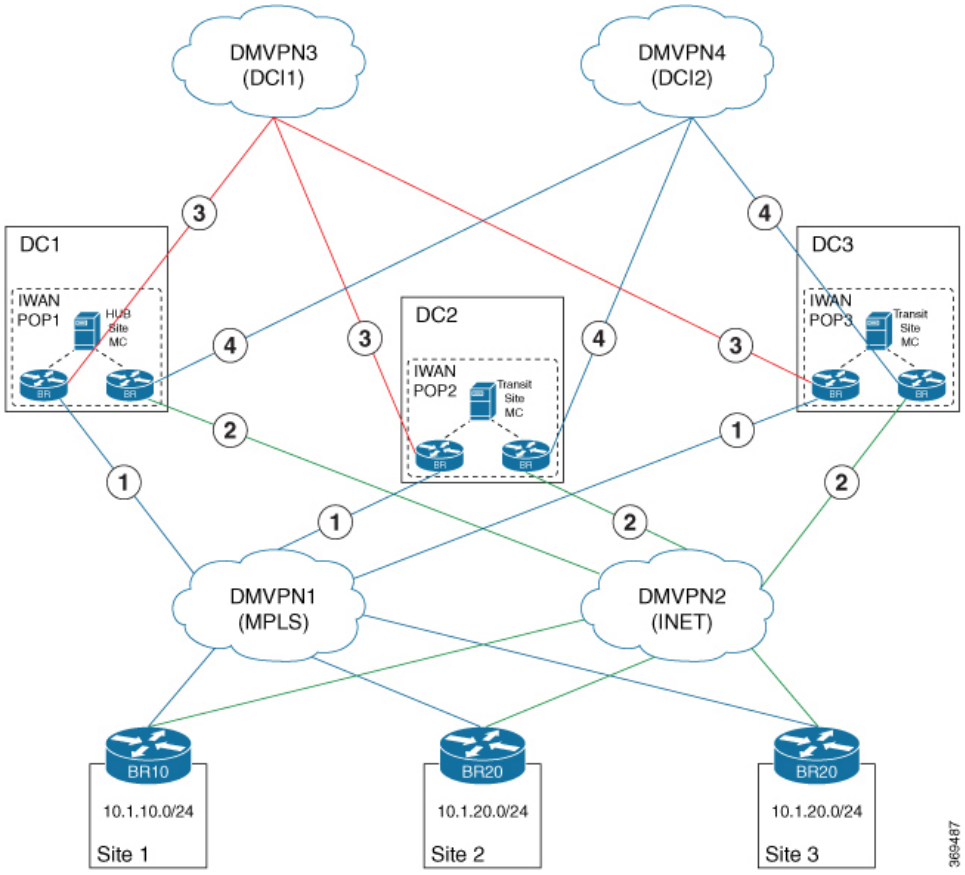
- The IDC feature must be enabled on both peer masters. It is recommended to use the same overlay routing protocol for all WAN and DCI tunnels.

Information About PfRv3-Inter-DC-Optimization

Datacenter Optimization

The following figure illustrates the PfRv3 Inter-DC Optimization feature where traffic between hub sites DC1, DC2 and DC3 are routed to forward specific traffic trough a specific hub. The figure shows four paths can be used as candidates for the traffic from DC1 to DC2. IDC1 and IDC2 are Inter-DC links those can be used for this traffic. MPLS and INET are normal WAN paths that can also be used for this traffic as candidates. It depends on the path-preference policy specific to inter-DC optimization.

Figure 240: Datacenter Optimization



The PfRv3 Inter-DC Optimization feature can be enabled with the **inter-dc** command in domain master controller advanced mode. All hubs in the network must be connected through WAN overlay or DCI overlay. All hub and transit hub masters must be enabled with this feature locally. WAN overlay is configured by defining a WAN interface using the **domain path** command. DCI overlay is configured by defining a DCI interface using the **domain dci-path** command.

The salient points of the PfRv3 Inter-DC Optimization feature are as follows:

- The **domain dci-path** command enables route control which routes the transit traffic on all DCI interfaces in ingress direction.
- Traffic classes are learnt based on the egress aggregate update and traffic channels over the WAN and DCI overlay.
- Tunnel addresses and path ID mapping are exchanged by site capability between the hub and transit masters.

Note The tunnel IP address for corresponding interface or path-id is advertised among the hub and transit masters when the PfRv3 Inter-DC Optimization feature is enabled.

DCI Path Options

Based on the actual deployment requirement, you can choose any of the following options for providing the DCI path:

Using the existing DMVPN overlay and the same tunnel interface:

In the hub to spoke DMVPN tunnel interface configuration, there is no dmvpn peer between DC sites. So, if the normal hub tunnel interface is used as DCI path, some additional configuration should be added to set up the dmvpn peer between DC sites, such as ip nhrp nhs 161.1.0.5 nbma 155.155.155.5 multicast above.

Using an independent DCI link(s) with independent DMVPN overlay

When there is dedicated DCI links between DC sites, a dedicated DMVPN overlay can be used as DCI path. And ideally, the dedicated DCI links are more stable than the normal WAN links. Using the existing hub to spoke DMVPN, or using a dedicated DCI DMVPN built over dedicated DCI links will depend on the available interfaces in the network, and which solution will meet the need of the network

Note A third option of building a second set of DMVPN tunnels using the same transport as the existing DMVPN hub and spoke network is not recommended and it has not been validated.

How to Configure PfRv3-Inter-DC-Optimization

Specifying the DCI interface on a Hub Site

```
enable
configure terminal
interface tunnel155
 domain dci-path DCI path-id 11
   exit
```

Configuring Inter-DC on Hub Master Controller

To configure the Inter-DC Optimization feature on the hub master controller, use the following commands:

```
enable
configure terminal
domain default
 vrf green
  master hub
   source-interface Loopback1
   site-prefixes prefix-list HUBPFX
   advanced
    inter-dc
     enterprise-prefix prefix-list ENTPFX
     class BUSINESS sequence 10
      match dscp ef policy custom
      priority 1 one-way-delay threshold 100
    interdc-path-preference DCI1 DCI2 fallback MPLS next-fallback INET
    exit
```

Configuring Inter-DC on Transit Hub

To configure Inter-DC on the transit hub, use the following commands:

```
enable
Configure terminal
domain default
 vrf green
  master transit 2
   source-interface Loopback1
   site-prefixes prefix-list HUBPFX
   hub 100.10.1.1
   advanced
inter-dc
  class BUSINESS sequence 10
    interdc-path-preference DCI1 fallback MPLS next-fallback INET
exit
```

Specifying IDC Local Policy

This is an optional task to overwrite the global path-preference.

```
enable
configure termainal
domain default
 vrf green
  master transit 2
   class BUSINESS sequence 10
    interdc-path-preference DCI1 fallback ISP1 next-fallback ISP2
exit
```

Verifying Inter-DC Configuration

```
HMCBR# show domain default vrf green master status
  *** Domain MC Status ***
 Master VRF: green
  Instance Type:    Hub
  Instance id:      1
  Operational status:  Up
  Configured status:  Up
  Loopback IP Address: 100.10.1.1
  Global Config Last Publish status: Peering Success
  Smart Probe Profile:
    General Monitor:
      Packets per burst: 1
      Interval(secs): 1
    Quick Monitor:
      Packets per burst: 20
      Interval(secs): 1
  Load Balancing:
   Admin Status: Disabled
   Operational Status: Down
   Enterprise top level prefixes configured: 0
  Route Control: Enabled
  Transit Site Affinity: Enabled
  Load Sharing: Enabled
  Path Pruning Depth: Disabled
  Inter-DC Optimization: Enabled
```

Verifying Master Controller Configuration

```
HMCBR# show domain default vrf green master status

  *** Domain MC Status ***
 Master VRF: green
  Instance Type:    Hub
  Instance id:      1
  Operational status:  Up
  Configured status:  Up
  Loopback IP Address: 100.10.1.1
  Global Config Last Publish status: Peering Success
..... .....
Minimum Packet Loss Calculation Threshold: 15 packets
  Minimum Bytes Loss Calculation Threshold: 1 bytes
  Borders:
    IP address: 100.10.3.1
    Version: 2
    Connection status: CONNECTED (Last Updated 15:44:28 ago )
    Interfaces configured:
      Name: Tunnel10 | type: external | Service Provider: ISP1 path-id:3 | Status: UP |
Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 0

      Name: Tunnel40 | type: external | Service Provider: ISP4 path-id:9 | Status: UP |
Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 0
      Name: Tunnel155 | type: DCI | Service Provider: DCI1 path-id:103 | Status: UP |
Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 0
    Tunnel if: Tunnel0
    IP address: 100.10.1.1
    Version: 2
    Connection status: CONNECTED (Last Updated 15:44:21 ago )
    Interfaces configured:
      Name: Tunnel10 | type: external | Service Provider: ISP1 path-id:1 | Status: UP |
Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 0
      Name: Tunnel30 | type: external | Service Provider: ISP3 path-id:7 | Status: UP |
Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 0
      Name: Tunnel155 | type: DCI | Service Provider: DCI1 path-id:101 | Status: UP |
Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 0
```

Verifying the Channel Status

```
HMCBR# show domain default vrf green master channels
   Legend: * (Value obtained from Network delay:)
Channel Id: 7  Dst Site-Id: 100.20.1.1  Link Name: ISP1  DSCP: default [0] pfr-label: 0:0
| 0:3 [0x3] TCs: 0  BackupTCs: 0
  Channel Created: 15:43:53 ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: FALSE
  Inter-DC Channel: FALSE
  Interface Id: 25
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Unreach Probing only: OFF
  Estimated Channel Egress Bandwidth: 0 Kbps
```

```
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
   ODE Statistics:
      Received: 0
   TCA Statistics:
      Received: 0 ; Processed: 0 ; Unreach_rcvd: 0 ; Local Unreach_rcvd: 0
      TCA lost byte rate: 0
      TCA lost packet rate: 0
      TCA one-way-delay: 0
      TCA network-delay: 0
      TCA jitter mean: 0
Channel Id: 117  Dst Site-Id: 100.16.1.1  Link Name: ISP1  DSCP: default [0] pfr-label:
3:13 | 0:3 [0x30D0003] TCs: 0  BackupTCs: 0
  Channel Created: 15:33:02 ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Inter-DC Channel: TRUE
  Interface Id: 25
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Unreach Probing only: OFF
  Estimated Channel Egress Bandwidth: 0 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    100.16.1.1/32 (Routable)
    100.16.0.0/16 (Routable)
    100.10.0.0/16 (Routable)
    100.15.0.0/16 (Routable)
    100.0.0.0/8 (Routable)
   ODE Statistics:
      Received: 0
   TCA Statistics:
      Received: 0 ; Processed: 0 ; Unreach_rcvd: 0 ; Local Unreach_rcvd: 0
      TCA lost byte rate: 0
      TCA lost packet rate: 0
      TCA one-way-delay: 0
      TCA network-delay: 0
      TCA jitter mean: 0
```

Example Configurations for PfRv3 Inter-DC

Example for Policy Configured on the Hub MC with Inter DC

In this example, the policy can work on the normal hub-spoke traffic and the IDC traffic. For IDC traffic, the ‘interdc-path-preference’ takes effect. DCI1 and DCI2 are primary paths. If they are out-of-policy, the MPLS, which is a backup path, will be used. For normal hub-spoke traffic, the ‘path-preference’ takes effect. The other configuration is same as normal PfRv3 policy. For example, the threshold of delay is 100 ms for both the IDC traffic and the normal hub-spoke traffic.

```
class BUSINESS sequence 10
    match dscp ef policy custom
    priority 1 one-way-delay threshold 100
    path-preference MPLS fallback INET
    interdc-path-preference DCI1 DCI2 fallback MPLS next-fallback INET
```

Example for Policy Configured on the Transit Hub MC with Inter DC

On transit hub master, you can see the same policy. But, if the **interdc-path-preference** is configured on this transit hub. The local interdc-path-preference will overwrite the policy from hub site

```
class BUSINESS sequence 10
    interdc-path-preference DCI1 fallback MPLS next-fallback INET
```

Example for 'show domain vrf master policy' on hub master

```
------------------------------------------------------------------------------
Global-policy-list:

  class BUSINESS sequence 10
    path-preference MPLS fallback INET
    class type: Dscp Based
      match dscp ef policy custom
        priority 1 one-way-delay threshold 100 msec
------------------------------------------------------------------------------
InterDC-policy-list:

  class BUSINESS sequence 10
interdc-path-preference DCI1 DCI2 fallback MPLS next-fallback INET
class type: Dscp Based
      match dscp ef policy custom
        priority 1 one-way-delay threshold 100 msec
```

Additional References for PfRv3-Inter-DC-Optimization

Related Documents

Related Topic	Document Title
Performance Routing commands	Cisco IOS Performance Routing Version 3 Command Reference

CHAPTER 287

Direct Cloud Access

The Direct Cloud Access IWAN 2.3 feature enables users at branch sites to have best application experience to SaaS applications, such as, Office 365, Google services, with reduced cost. This feature helps in constantly monitoring network and application performance and select the optimized paths (usually local break out from branch to Cloud SaaS applications instead of back-haul to the data center). Non-SaaS traffic still back-haul to data center for further inspection.

Feature Information for Configuring Direct Cloud Access

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 344: Feature Information for Direct Cloud Access IWAN 2.3

Feature Name	Releases	Feature Information
Direct Cloud Access IWAN 2.3	Cisco IOS XE Fuji 16.8.1	The Direct Cloud Access (DCA) feature allows traffic from trusted applications, part of well-trusted domains, to pass the local Internet security check because traffic from these trusted applications have a lower security risk than untrusted Internet sites. The following commands were introduced or modified: **domain path**, **path-preference**, **show domain dca-status**, **show domain default border**, **show domain default policy**, **show domain vrf border channels**, **show domain vrf master channels**.

Prerequisites for Configuring Direct Cloud Access

- NAT:

  To enable a host that typically operates in a private network directly communicate with a SaaS application in a public network, use a NAT. Enable NAT on the same router that has DCA enabled or other devices in the path.

- Firewall security:

  To improve security, you can enable a firewall, such as a zone-based firewall (ZBFW), in the path.

Note By default OpenDNS is used as DNS resolver for SaaS traffic, but you can choose to use other DNS resolver such as Google DNS resolver 8.8.8.8. OpenDNS license/registration is not a must if you don't need OpenDNS security services.

Restrictions for Configuring Direct Cloud Access

- IPv6 address is not supported.
- DCA is not supported if the DNS traffic does not pass through the router which is enabled with DCA.
- DCA does not work if SaaS applications use proxy. All traffic going to proxy server as DCA may not classify these applications and cannot perform local breakout for traffic that is bound to proxy.
- Applications that directly access the content and not through DNS resolution, NBAR may fail to classify as SaaS and cannot provide local break-out.
- DCA may not work on a device when NBAR classification results are not available on the device. You must customize NBAR to classify the results to support DCA.
- This feature depends on applications classification. SD-AVC helps in better classification with NBAR.
- To access SaaS applications, a public IP address is required. NAT helps translate the user’s private IP address to a public IP address. Configure NAT on the border router that has DCA enabled, or on other internet-facing devices.

Information About Configuring Direct Cloud Access

Direct Cloud Access Overview

The infrastructure of cloud-hosted services, such as Microsoft Office 365 and Google Apps, is in the cloud. Back-hauling traffic from remote users and sites through the private WAN to the data center via Internet imposes additional bandwidth requirements on the private WAN and may add latency to each connection. Moreover, private WAN connectivity is more expensive than direct Internet connections, which could add a tremendous amount of cost to the equation.

The Direct Cloud Access IWAN 2.3 feature implements direct cloud access (DCA) on Cisco IWAN networks and allows trusted SaaS traffic to be forwarded out over the optimized path (directly local break out) while other traffic still back-haul to headquarters over VPN. DCA monitors the candidate path (DCA path, back-haul path to headquarter) performance and chooses the optimized path in policy to get the best SaaS application performance. While adding direct Internet connectivity to the branch site without back hauling to data center, IWAN DCA provides the security capability at branch site by enabling security features like NAT and Firewall (Zone-based Firewall, Snort IPS, etc.) at branch sites.

Features

DCA features include:

- Automatic configuration of Cisco Umbrella Connector (supported from Cisco IOS XE Gibraltar 16.10.1)
- Support for policy configured on a centralized hub, or per-site customized local policy
  - Customized local policy overrides global plicy.
  - If a hub connection goes down, local policy remains in effect.
- Support for P2P interface, such as dialer interface, as DCA interface

Benefits of Direct Cloud Acces

- Reduced operation cost as SaaS traffic no longer needs to go to headquarters which consume additional headquarter network bandwidth.
- Business processes run faster through direct network access to the major cloud providers. A traffic classification mechanism is required in order to achieve direct Internet access for selected cloud applications.

Direct Cloud Access Architecture

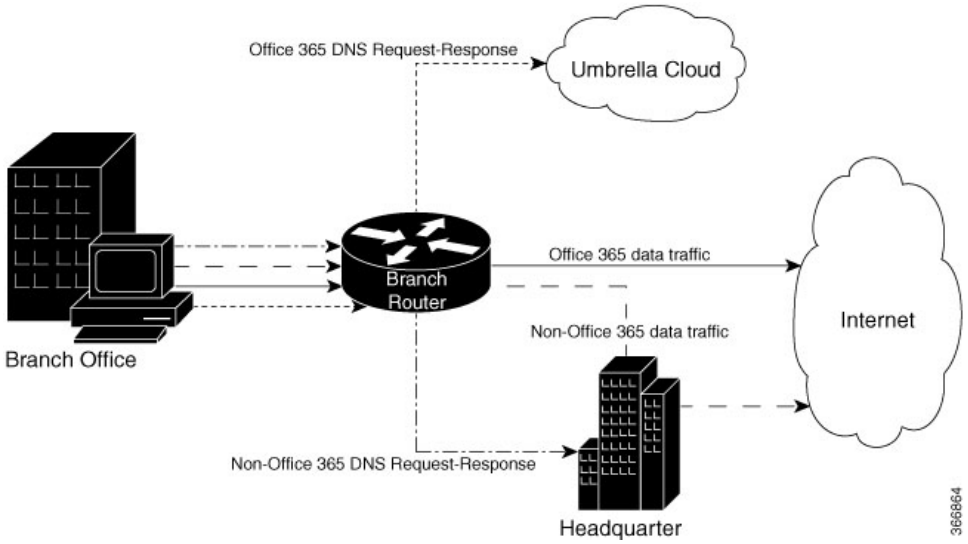
The overlay DMVPN WAN tunnels on a branch router are configured to dynamically learn the service provider they are connected to. An underlay interface is identified as a direct access interface via configuration.

Packets from the LAN side on a branch site are sent over the overlay when packets do not match the criteria of the configured application. When a flow matches the DCA criteria, the packets are directed to the DCA interface that is specified in the path preference. DCA interfaces can be listed in the order of priority in the path preference configuration of the policy for the application. The DCA interfaces are evaluated in the order of the configured path preference priority.

NBAR classification occurs at LAN ingress. NBAR provides the application ID, which is exported by the border router. If a match occurs on the Master Controller for an application, the policy for the application is applied to the traffic class for the specific flow.

The following figure explains the DCA functionality for Office365 application:

Figure 241: DCA for Office365



The following actions are performed to achieve DCA functionality:

- Classify all the cloud applications based on the DNS.
- Intercept DNS traffic and make decisions based on the classification.
  - If the traffic is from a trusted application, direct Internet access is provided. Ensure that security concerns are addressed for the breakout traffic, which include, constant application monitoring, choosing network performance over candidate paths (DCA path, back-haul path), selecting the optimized path according to policy (if DCA path is not good), back-hauling SaaS traffic to data center and reverting back if DCA path recovered.
  - If the traffic is not from a trusted application, the traffic is passed it to the Headquarter for further security inspection and processing.
- Route HTTP, HTTPS data traffic to Internet or Headquarter depending on the above decision.

Designate an Underlay Interface as Direct Access Interface

An interface of the border router must be designated as direct access interface. **domain path** *path-name* **direct-cloud-access** command to specify the direct access interface. A service provider may have multiple links of direct access and each of the direct access interface is measured independently.

When an interface is selected to be the direct access interface, all traffic to the whitelisted applications is directed through the direct access interface. If there are multiple direct access interfaces, the traffic is directed on one direct access interface depending on the performance metrics and policy.

Direct Cloud Access Components

Direct Cloud Access functionality has the following components:

Cisco Umbrella Connector

To achieve location proximity, the SaaS server must be closer to the branch router to achieve better application performance. Generally, DNS requests for a SaaS application are destined to an enterprise DNS resolver. However, the DNS request must be changed from enterprise DNS resolver to a public DNS resolver, such as, OpenDNS resolver or Google DNS resolver. The public DNS resolver helps in placing the SaaS server closer to the branch router by using Cisco Umbrella connector. OpenDNS account and registration is not mandatory.

DNS requests must be unencrypted traffic from the endpoint to the DNS server. Each direct access interface must be configured with Open DNS.

NBAR Classification

Network Based Application Recognition (NBAR) is a classification engine that recognizes and classifies a wide variety of protocols and applications. NBAR uses several classification information metadata such as application name, ID, traffic class, business relevance, and so on.

For Direct Cloud Access functionality, once NBAR recognizes the DNS traffic as belonging to interesting cloud application, it attaches this information to DNS packet in a way so that the umbrella connector feature can extract and use the information.

Cisco NBAR provides the first packet classification for some applications. Cisco NBAR uses DNS learning for application recognition of user defined and predefined domains, Once the server is learned from the DNS response, traffic going to this server can be classified as FIFO. SD-AVC also improves the first packet classification result.

Performance Routing Version 3

Performance Routing version 3 (PfRv3) delivers intelligent path control for application-aware routing across the WAN. Once a DNS response is received, the data traffic (HTTP, HTTPS etc.) from cloud application is provided direct Internet access (local break-out) or is sent to the headquarter for further security inspection.

IPSLA

IPSLA is enabled automatically by PfRv3 to probe each SaaS application over candidate paths by using IPSLA HTTP operation. PfRv3 leverages the metrics reported by IPSLA to select the optimized path.

SaaS Reachability and Performance Management

Performance and reachability of each whitelisted application determines the path that an application takes. PfR measures the reachability and performance of all VRFs and enables and shares one measurement across multiple VRFs.

Next-Hop Reachability

One DSCP-agnostic channel is created as the next-hop for the direct access interface. The DSCP of DCA channel is configured as FF. The routing protocol configured on the direct access interface determines the next hop for the channel.

Performance Measurement

After the channel next hop is up, the service is reached via next hop by using the following steps:

Application Domain Mapping

Application to domain URL and Differentiated Services Code Point (DSCP) mapping must be configured on the master controller of each branch router so that IPSLA can measure the SaaS application using right domain and DSCP.

Reachability and Performance Probing

Measuring network characteristics is performed using IPSLA. IPSLA probes are not sent per VRF, instead, PFR creates a probing layer for all the VRFs and path preferences in the VRFs in a domain. Reachability and performance can be verified per application by using the **show domain** *domain-name* **border dca** command. This command provides information per application, per interface for a border router.

Traffic Steering and Flow Stickiness

When DCA is implemented on a network, traffic classes are automatically created for interested applications. The applications configured in the policy includes path preferences, which corresponds to the respective DSCP configured per application.

When selecting a path, PfR assigns a path to a flow that is destined to a service, for example, Offic365. These flows might traverse a NAT device or a firewall device that maintains the state for the flow sequence numbers. Changing the flow during packet traversal may lead to flow reset. Therefore, when a path is selected, flows must align to that path only. If a path is unreachable, the flow is reset by the client and retried. If the path experiences packet loss but still usable, new flows are routed via alternate paths.

Local Policy Configuration

Direct Cloud Access (DCA) policy can be configured on a centralized hub, or it can be configured on any individual site as a customized local policy. To configure local DCA policy, use the **policy local type DCA** command.

- Customized local policy overrides global policy.
- If a hub connection goes down, local policy remains in effect.

Example of Local Policy Configuration

```
policy local type DCA
  class DCA sequence 4
    match application ms-cloud-group saas-dca
      path-preference DCA1 fallback DCA2
```

How to Configure Direct Cloud Access

Assign an Underlay Interface as Direct Access Interface

The following configuration snippet explains how to assign an Ethernet interface as direct access interface.

```
Router(config)# interface Ethernet 0/1
Router(config-interface)# domain path ATT-DCA direct-cloud-access
```

Define PfR Policy for SaaS Application on Hub Master Controller

The following configuration snippet explains how SaaS application policies are defined on hub master controller at a central point and published to all branch sites. There is no need to define policies at each branch sites because branch sites still have the capability to customize the interested SaaS.

```
Router(config)# domain iwan Router
Router(config-domain)# vrf green
Router(config-domain-vrf)# master hub
Router(config-domain-vrf-master)# class BUSINESS-CRITICAL sequence 10
Router(config-domain-vrf-master-class)# match app-group ms-cloud-group policy custom
Router(config-domain-vrf-master-class-match)# priority 1 delay 500 ms
Router(config-domain-vrf-master-class-match)# exit
Router(config-domain-vrf-master-class)# path-preference ATT-DCA fallback ATT next-fallback
 INET
```

Define SaaS Application Mapping on Branch Master Controller

To measure the SasS application's reachability and performance, the domain URL and DSCP must be specified for IPSLA probing for each SaaS application.

Use HTTP ping to probe a specific SaaS to determine reachability and performance. The system has built-in default URL domains for popular SaaS applications. For a complete list, use **show domain xxx master dca domain-map**.

Note If there are multiple VRFs, IP SLA probing is performed for all domains defined for each VRF and the same IP SLA ID is used for each domain group in the VRF.

If a desired SaaS is not included in the list, create a domain map for the service in PfRv3. For example, to add Servicenow:

```
master branch
 domain-map
 application servicenow-group domain http://www.servicenow.com dscp af21
```

Configure a DNS Resolver

By default, DNS requests for white-listed SaaS are intercepted by Umbrella, and the OpenDNS resolver is used to achieve location proximity.

Optionally, configure a specific DNS resolver, either on a hub master controller or on an specific branch master controller. Configuring a DNS resolver on a specific branch overrides, for that branch, the DNS resolver configured on the hub.

Hub

Use the following on a hub master controller to configure a DNS resolver for all DCA branches.

```
domain default
  master hub
    advanced
      dns-redirect dns-server-address
```

Example:

```
domain default
  master hub
    advanced
      dns-redirect 8.8.8.8
```

Branch

Use the following on a branch master controller to configure a DNS resolver for the branch, overriding the hub setting.

```
domain default
  master branch
    dns-redirect dns-server-address
```

Example:

```
domain default
  master branch
    dns-redirect 8.8.8.8
```

Configure the HTTP Ping Probe Interval

The HTTP ping probe uses a default probe interval of 30 seconds.

Optionally, you can configure a specific interval on the hub master controller, which applies the change to all DCA branches, or to a branch master controller, to change the interval for a specific branch.

Hub

Use the following on a hub master controller to configure the interval for all DCA branches.

```
domain default
  master hub
    advanced
      dca-probe-http-interval interval-in-seconds
```

Example:

```
domain default
  master hub
    advanced
      dca-probe-http-interval 20
```

Branch

Use the following on a branch master controller to configure the interval for a specific branch. The branch setting overrides a setting made at the hub.

```
domain default
  master branch
    dca-probe-http-interval interval-in-seconds
```

Example:

```
domain default
  master branch
    dca-probe-http-interval 20
```

Verify and Monitor Direct Cloud Access Configuration

Use the following commands to verify and monitor DCA configuration.

- **show domain iwan master traffic-classes summary**
- **show domain iwan master traffic-classes detail**
- **show domain iwan master traffic-classes dca detail**
- **show domain iwan master traffic-classes dca application**
- **show domain** *domain-name* **border dca**

  Displays information about reachability and metrics collected for all paths towards a service. This command helps in understanding the behavior of various paths for a service and how PFR is selecting the best paths depending on the metrics.

```
Device# show domain iwan border dca

[*] PFR created IP SLA entry ID
IPSLA DNS Resolver:208.67.220.220

App    DSCP  RTT/ms  DCA2   MPLS1
        thresh  Gi0/0/2   Tu10 (0:1)
           RTT/ms[*]   RTT/ms[*]
share-point    default  1000  7 [31]   --
youtube    default  1000  78 [33]   --
box    default  1000  7 [39]   --
dropbox    default  1000  3 [41]   --
google-services default     1000  108 [49]   --
google-group    default  1000  109 [51]   --
gtalk-group    default  1000  112 [53]   --
hangouts-group    default  1000  115 [55]   --
ms-lync-group    default  1000  6 [57]   --
ms-cloud-group    default  1000  7 [59]   --
```

- **show domain default policy**

  Displays the default policy on the master controller.

```
Device# show domain default master policy
       No Policy publish pending
--------------------------------------------------------------------------------
  class SOCIAL-NETWORKING sequence  11
        class type: Application Based
             match application skype policy custom
                   priority 1 delay threshold 500 msec
```

- To troubleshoot, use **debug domain default master dca** and *debug domain default border dca* commands.

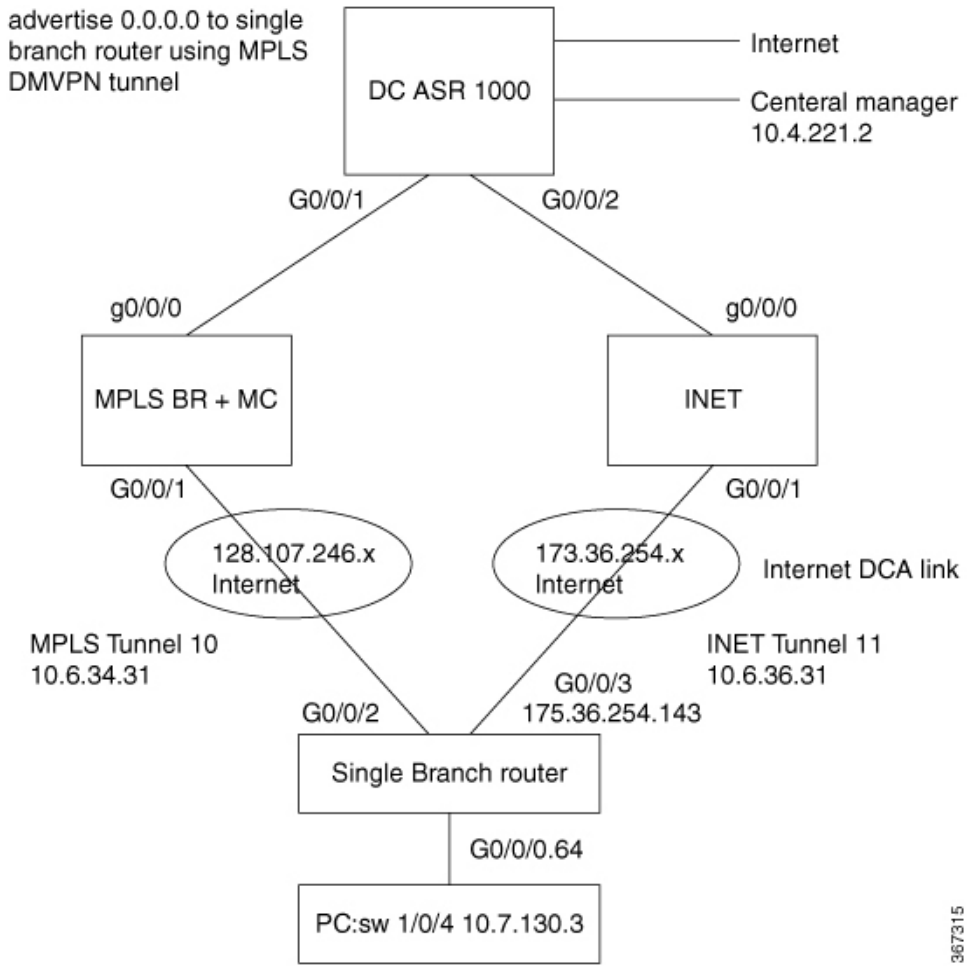
Configuration Examples for Configuring Direct Cloud Access

Example: Configure DCA Link on a Single Branch Router

Overview

In this example, DCA is configured on Cisco IWAN network with a single branch router as shown in the following topology.

Figure 242: DCA Link on a Single Branch Router



Umbrella Service

Beginning with Cisco IOS XE Gibraltar 16.10.1, the Umbrella service configuration is automatic.

Underlay Interface

DCA is configured on WAN underlay interface in order to distinguish tunnel WAN interface.

```
interface GigabitEthernet0/0/3  ! INET branch WAN DCA interface
domain iwan path DCA1 direct-cloud-access
umbrella out
```

Optionally, a second DCA can be created as WAN underlay interface.

```
interface GigabitEthernet0/0/2  ! INET branch DCA2 interface
 domain iwan path DCA2 direct-cloud-access
 umbrella out
```

Create Domain Map

Optionally, create a domain map for a specific SaaS not included by default.

```
master branch
 domain-map
 application servicenow-group domain http://www.servicenow.com dscp af21
```

Hub Master Controller Configuration

The policy can be local or from a centralized hub master controller. Configure a hub master controller as follows:

Note Configure only one master controller, either at a hub site or a branch site.

```
domain default
vrf default
  master hub

  class DCA sequence 4
    match application ms-cloud-group saas-dca
    path-preference DCA1 fallback DCA2
```

Branch 1 and Master Controller Configuration

A branch site can serve as master controller instead of a hub site. In this example, Branch 1 serves as master conroller. The configuration includes LAN interface and WAN (DCA) interface.

Note Configure only one master controller, either at a hub site or a branch site.

```
domain default
  vrf default
    border
      master local
    master branch
      source-interface Loopback0
      hub 100.20.1.1
```

LAN interface configuration:

```
interface GigabitEthernet3.30
  description B1MCBR-LAN
  encapsulation dot1Q 30
  ip address 10.20.0.1 255.255.255.0
  ip nat inside
```

Assigning the DCA to a WAN interface:

```
interface GigabitEthernet2.30
  encapsulation dot1Q 30
    ip vrf forwarding fvrf
    ip address 10.20.1.1 255.255.255.0
    ip nat outside
    domain path DCA1 direct-cloud-access
```

Branch 2 Configuration

This branch configuration includes LAN interface and WAN (DCA) interface.

```
domain default
  vrf default
     border
       source-interface Loopback0
       master 192.168.3.22
```

LAN interface configuration:

```
interface GigabitEthernet3.30
  description B1MCBR-LAN
  encapsulation dot1Q 30
  ip address 10.20.0.1 255.255.255.0
  ip nat inside
```

Assigning the DCA to a WAN interface:

```
interface GigabitEthernet2.30
  encapsulation dot1Q 30
    ip vrf forwarding fvrf
    ip address 10.20.1.1 255.255.255.0
    ip nat outside
    domain path DCA2 direct-cloud-access
```

Verifying the Configuration

The following commands are used to verify the configuration. To verify OpenDNS configuration, use the **show umbrella deviceid** and **show umbrella configuration** commands.

```
router# show umbrella deviceid

Device registration details
Interface Name                    Tag                   Status                 Device-id
GigabitEthernet3.64     inside-network   200 SUCCESS     010a3d458c172b8b

router# show umbrella configuration
Umbrella Configuration
========================
   Token: 7772166EF2E473ADE8FA2204B37D0BD7001FE4F5
   OrganizationID: 2090229
   Local Domain Regex parameter-map name: dns_bypass
   DNSCrypt: Enabled
   Public-key: B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79

   UDP Timeout: 5 seconds
   Resolver address:
       1. 208.67.220.220
       2. 208.67.222.222
       3. 2620:119:53::53
```

```
       4. 2620:119:35::35
Umbrella Interface Config:
       Number of interfaces with "umbrella out" config: 2
         1. GigabitEthernet0/0/0
             Mode     :  OUT
             VRF      : IWAN-TRANSPORT-2(Id: 3)
         2. GigabitEthernet0/0/1
             Mode     :  OUT
             VRF      : IWAN-TRANSPORT-1(Id: 2)
       Number of interfaces with "umbrella in" config: 1
         1. GigabitEthernet3.64
             Mode       : IN
             DCA        : Enabled
             Policy Name: umbrella-direct-access
             Tag        : lan064
             Device-id  : 010a3d458c172b8b
             VRF        : global(Id: 0)
```

To verify the DCA configuration, use the following commands:

- **show domain iwan border dca**
- **show domain iwan master dca status**
- **show domain iwan master traffic-classes summary**
- **show domain iwan master traffic-classes detail**
- **show ip sla summary**
- **show ip sla configuration**
- **show ip sla statistics**
- **show flow monitor name flow-monitor cache format table**

Use the **show ip sla summary**, **show ip sla configuration**, and **show ip sla statistics** commands to verify the probe functions.

Use the **show flow monitor** command to verify that the flow is passes through the DCA path.

```
Router# show domain iwan master traffic-classes summary

APP - APPLICATION, TC-ID - TRAFFIC-CLASS-ID, APP-ID - APPLICATION-ID
Current-EXIT - Service-Provider(PFR-label)/Border/Interface(Channel-ID)
UC - UNCONTROLLED, PE - PICK-EXIT, CN - CONTROLLED, UK - UNKNOWN

Dst-Site-Pfx      Dst-Site-Id       State DSCP        TC-ID      APP-ID    APP
Current-Exit

DCA               Internet          CN    default[0]  30         9424      ms-cloud-group
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
DCA               Internet          CN    default[0]  29         4478      ms-lync-group
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
DCA               Internet          CN    default[0]  28         8388      hangouts-group
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
DCA               Internet          CN    default[0]  27         4692      gtalk-group
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
DCA               Internet          CN    default[0]  26         4456      google-group
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
DCA               Internet          CN    default[0]  25         218104328 google-service
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
DCA               Internet          CN    default[0]  21         50349148  dropbox
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
```

```
DCA               Internet          CN    default[0]  20         218104882 box
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)
DCA               Internet          CN    default[0]  17         218103890 youtube
DCA1(0:0|0:0)/10.255.241.31/Gi0/0/3(Ch:6)

Router# show domain iwan border dca

[*] PFR created IP SLA entry ID
IPSLA DNS Resolver:208.67.220.220

App    DSCP  RTT/ms  DCA2   MPLS1
        thresh  Gi0/0/2   Tu10 (0:1)
           RTT/ms[*]   RTT/ms[*]
share-point    default  1000  7 [31]   --
youtube    default  1000  78 [33]   --
box    default  1000  7 [39]   --
dropbox    default  1000  3 [41]   --
google-services default     1000  108 [49]   --
google-group    default  1000  109 [51]   --
gtalk-group    default  1000  112 [53]   --
hangouts-group    default  1000  115 [55]   --
ms-lync-group    default  1000  6 [57]   --
ms-cloud-group    default  1000  7 [59]   --

Router# show domain iwan master traffic-classes detail

 Dst-Site-Prefix: DCA               Application: ms-cloud-group   DSCP: default [0] Traffic
 class id:30 app_id:9424
  Clock Time:                 22:13:32 (UTC) 01/17/2018
  TC Learned:                 4d23h ago
  Present State:              CONTROLLED
  Current Performance Status: not monitored (internet)
  Current Service Provider:   DCA1 since 4d23h
  Previous Service Provider:  Unknown
  BW Used:                    0 bps
  Present WAN interface:      GigabitEthernet0/0/3 in Border 10.255.241.31
  Present Channel (primary):  6 DCA1 pfr-label:0:0 | 0:0 [0x0]
  Backup Channel:             4 DCA2 pfr-label:0:0 | 0:0 [0x0]
  Destination Site ID:        Internet
  DNS Primary Channel:        6 DCA1 pfr-label:0:0 | 0:0 [0x0]
  DNS Backup Channel:         4 DCA2 pfr-label:0:0 | 0:0 [0x0]
  Class-Sequence in use:      55
  Class Name:                 saasapp using policy User-defined
    priority 1 one-way-delay threshold 500 msec
  BW Updated:                 - ago
  Method for choosing channel: Random
  Reason for Latest Route Change:    Uncontrolled to Controlled Transition
  Route Change History

Router# show ip sla sum

ID           Type        Destination       Stats       Return      Last
                                                       Code        Run
---------------------------------------------------------------------------

*1255        http        216.58.217.164    RTT=198     OK          30 seconds ago
*1256        http        216.58.217.164    RTT=184     OK          30 seconds ago
*1257        http        216.58.217.164    RTT=219     OK          30 seconds ago

*1258        http        216.58.217.164    RTT=219     OK          30 seconds ago

*1259        http        13.107.7.190      RTT=76      Http Error  30 seconds ago
```

```
Router# show ip sla config 1255

Entry number: 1255
Type of operation to perform: http
Target address/Source address: 216.58.217.164/172.16.1.1
Target port/Source port: 80/0
Type Of Service parameters: 0x0
Vrf Name: IWAN-TRANSPORT-2
HTTP Operation: get
HTTP Server Version: 1.0
URL: http://www.google.com
Proxy:
Raw String(s):
Cache Control: enable
Owner:
Tag:
Operation timeout (milliseconds): 30000
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 20000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None

Router# show ip sla statistics 1255

IPSLA operation id: 1255
        Latest RTT: 179 milliseconds
Latest operation start time: 19:09:14 UTC Fri Jan 26 2018
Latest operation return code: OK
Latest DNS RTT: 6 ms
Latest TCP Connection RTT: 62 ms
Latest HTTP Transaction RTT: 111 ms
Number of successes: 29
Number of failures: 0
Operation time to live: Forever

Router# show flow monitor Monitor-FNF-IWAN cache format table | i office
54.209.129.73    172.31.1.2                80          62102  Gi0/0/0               Input
            6  layer7 ms-office-365             0.0.0.0                         /30  Null
                          4                   0x00
172.31.1.2       52.109.2.14              5110           443  Gi0/0/2.101           Output
           6  layer7 ms-office-web-apps         172.31.1.1                      /0  Gi0/0/0
                          7                   0x00
10.30.32.200     104.91.217.163           50319           443  Gi0/0/2.101           Input
            6  layer7 ms-office-365             172.31.1.1                      /0  Gi0/0/0
                          9                   0x00
172.31.1.2       208.67.222.222           52262            53  Null                  Output
          17  layer7 ms-office-365             172.31.1.1                     /30  Gi0/0/0
                          1  www.office.com   0x00
10.30.32.200     104.91.188.182           50341           443  Gi0/0/2.101           Input
            6  layer7 ms-office-365             172.31.1.1                      /0  Gi0/0/0
```

```
                    11                    0x00
10.30.32.200     104.91.188.182          50310            443  Gi0/0/2.101           Input
           6  layer7 ms-office-365              172.31.1.1                     /0  Gi0/0/0
                    11                    0x00
172.31.1.2       52.109.2.14              5108            443  Gi0/0/2.101           Output
           6  layer7 ms-office-web-apps         172.31.1.1                     /0  Gi0/0/0
                    8                     0x00
```

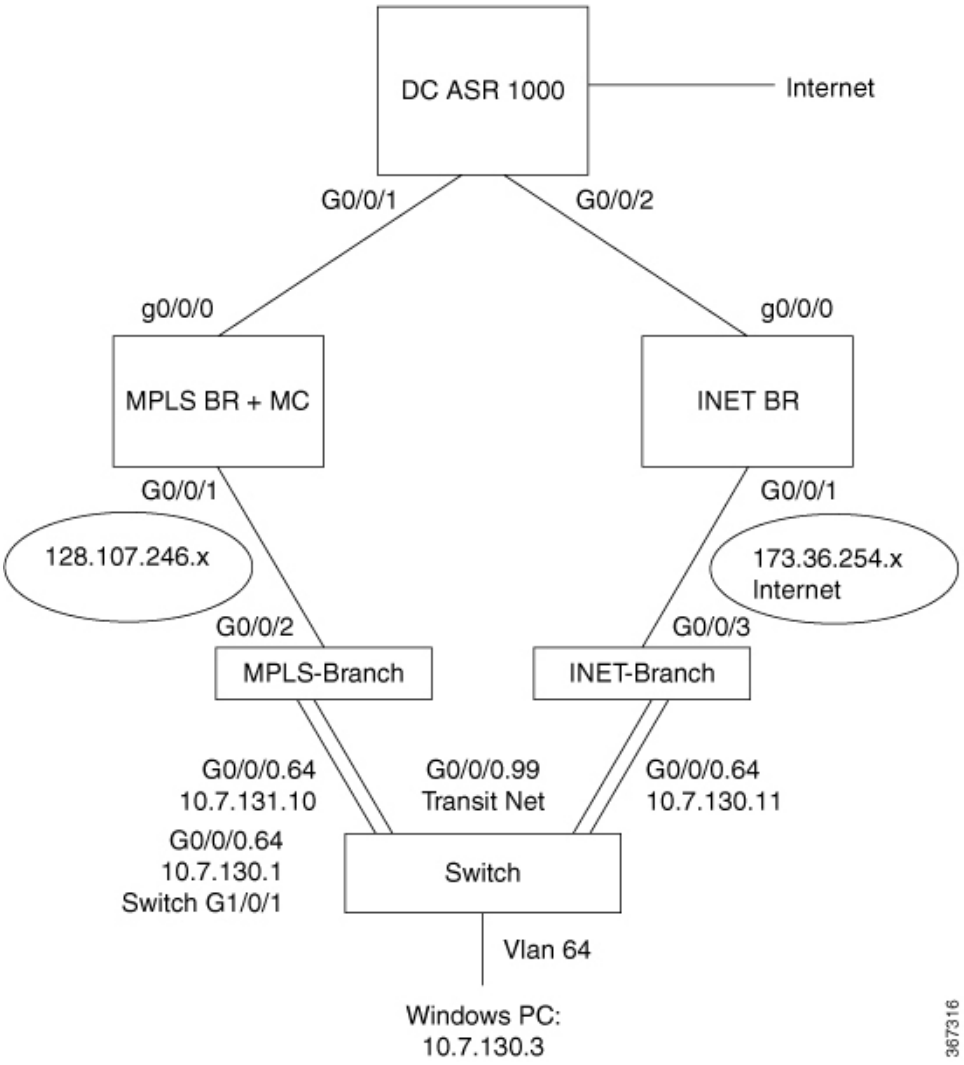
Example: Configure DCA Link on a Dual Branch Router

Overview

In this example, DCA is configured on Cisco IWAN network with a dual branch router as shown in the following topology.

The policy can be local or from a centralized hub. This example illustrates the use of a local policy in a non-IWAN scenario.

Figure 243: DCA Link on a Dual Branch Router



Branch 1 and Master Controller Configuration

A branch site can serve as master controller instead of a hub site. In this example, Branch 1 serves as master conroller. The configuration includes LAN interface and WAN (DCA) interface.

Note Configure only one master controller, either at a hub site or a branch site.

```
domain default
  vrf default
    border
      master local
    master branch
      source-interface Loopback0
            hub l00.20.1.1

  policy local type DCA
    class DCA sequence 4
      match application ms-cloud-group saas-dca
        path-preference DCA1 fallback DCA2
```

LAN interface configuration:

```
interface GigabitEthernet3.30
  description B1MCBR-LAN
  encapsulation dot1Q 30
  ip address 10.20.0.1 255.255.255.0
  ip nat inside
```

Assigning the DCA to a WAN interface:

```
interface GigabitEthernet2.30
  encapsulation dot1Q 30
    ip vrf forwarding fvrf
    ip address 10.20.1.1 255.255.255.0
    ip nat outside
    domain path DCA1 direct-cloud-access
```

Branch 2 Configuration

This branch configuration includes LAN interface and WAN (DCA) interface.

```
domain default
  vrf default
    border
      source-interface Loopback0
      master 192.168.3.22
```

LAN interface configuration:

```
interface GigabitEthernet3.30
  description B1MCBR-LAN
  encapsulation dot1Q 30
  ip address 10.20.0.1 255.255.255.0
  ip nat inside
```

Assigning the DCA to a WAN interface:

```
interface GigabitEthernet2.30
  encapsulation dot1Q 30
  ip vrf forwarding fvrf
  ip address 10.20.1.1 255.255.255.0
  ip nat outside
  domain path DCA2 direct-cloud-access
```

Example: Configuring Umbrella Branch for OpenDNS

Overview

Beginning with Cisco IOS XE Gibraltar 16.10.1, DCA configures the Cisco Umbrella Connector automatically on the router. However, it is still possible to configure Umbrella manually.

For example, if it is necessary to validate OpenDNS, you must configure Cisco Umbrella Connector on the branch, as shown in the example below.

Procedure

1. Configure the DNS server, setting the router's clock and time zone correctly.

```
ip domain name cisco.com
ip host api.opendns.com 67.215.92.210
```

2. Log into the OpenDNS portal to get an API token.

   https://login.opendns.com/?return_to=https://dashboard2.opendns.com

3. Import the certificate, entering a PEM-formatted CA certificate.

   (config)#**crypto pki trustpool import terminal**

   Enter a PEM-formatted CA certificate.

```
(config)#crypto pki trustpool import terminal
-----BEGIN CERTIFICATE-----
MIIElDCCA3ygAwIBAgIQAf2j627KdciIQ4tyS8+8kTANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBD
QTAeFw0xMzAzMDgxMjAwMDBaFw0yMzAzMDgxMjAwMDBaME0xCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxJzAlBgNVBAMTHkRpZ2lDZXJ0IFNIQTIg
U2VjdXJlIFNlcnZlciBDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
ANyuWJBNwcQwFZA1W248ghX1LFy949v/cUP6ZCWA1O4Yok3wZtAKc24RmDYXZK83
nf36QYSvx6+M/hpzTc8zl5CilodTgyu5pnVILR1WN3vaMTIa16yrBvSqXUu3R0bd
KpPDkC55gIDvEwRqFDu1m5K+wgdlTvza/P96rtxcflUxDOg5B6TXvi/TC2rSsd9f
/ld0Uzs1gN2ujkSYs58O09rg1/RrKatEp0tYhG2SS4HD2nOLEpdIkARFdRrdNzGX
kujNVA075ME/OV4uuPNcfhCOhkEAjUVmR7ChZc6gqikJTvOX6+guqw9ypzAO+sf0
/RR3w6RbKFfCs/mC/bdFWJsCAwEAAaOCAVowggFWMBIGA1UdEwEB/wQIMAYBAf8C
AQAwDgYDVR0PAQH/BAQDAgGGMDQGCCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYYY
aHR0cDovL29jc3AuZGlnaWNlcnQuY29tMHsGA1UdHwR0MHIwN6A1oDOGMWh0dHA6
Ly9jcmwzLmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydEdsb2JhbFJvb3RDQS5jcmwwN6A1
oDOGMWh0dHA6Ly9jcmw0LmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydEdsb2JhbFJvb3RD
QS5jcmwwPQYDVR0gBDYwNDAyBgRVHSAAMCowKAYIKwYBBQUHAgEWHGh0dHBzOi8v
d3d3LmRpZ2ljZXJ0LmNvbS9DUFMwHQYDVR0OBBYEFA+AYRyCMWHVLyjnjUY4tCzh
xtniMB8GA1UdIwQYMBaAFAPeUDVW0Uy7ZvCj4hsbw5eyPdFVMA0GCSqGSIb3DQEB
CwUAA4IBAQAjPt9L0jFCpbZ+QlwaRMxp0Wi0XUvgBCFsS+JtzLHgl4+mUwnNqipl
5TlPHoOlblyYoiQm5vuh7ZPHLgLGTUq/sELfeNqzqPlt/yGFUzZgTHbO7Djc1lGA
8MXW5dRNJ2Srm8c+cftIl7gzbckTB+6WohsYFfZcTEDts8Ls/3HB40f/1LkAtDdC
2iDJ6m6K7hQGrn2iWZiIqBtvLfTyyRRfJs8sjX7tN8Cp1Tm5gr8ZDOo0rwAhaPit
c+LJMto4JQtV05od8GiG7S5BNO98pVAdvzr508EIDObtHopYJeS4d60tbvVS3bR0
j6tJLp07kzQoH3jOlOrHvdPJbRzeXDLz-----END CERTIFICATE-----
```

Note This is the PEM-formatted version. Keep the "END CERTIFICATE" portion. Without this, the certificate will be lost after a router reloads.

4. (Optional) Configure local domains.

DNS queries directed to a local domain will remain untouched and will not be redirected to OpenDNS cloud.

```
parameter-map type regex dns_bypass
pattern www.cisco.com
pattern .*eisg.cisco.*
```

5. Configure the token.

```
parameter-map type umbrella global
token 0F32C32FEC26991C2B562D3C7FF844E0001C70E7
local-domain dns_bypass
```

Note Cisco OpenDNS is used by default. To use a different DNS resolver, add the following line:

resolver ipv4 *DNS-server- IP*

Note Enter a fake token for this step if you are using another DNS server or do not want to register with OpenDNS server.

6. Assign the above policy to a LAN interface.

```
GigabitEthernet0/0/0.100  ! INET branch LAN interface
umbrella in direct-cloud-access default lan100
```

7. Assign the Umbrella to a WAN interface.

```
interface GigabitEthernet0/0/3  ! INET branch WAN DCA interface
domain iwan path DCA1 direct-cloud-access
umbrella out
```

8. Apply **umbrella out** on all DCA interfaces. This includes the MPLS branch of a dual branch if the MPLS branch has a DCA interface.

```
interface GigabitEthernet0/0/2  ! MPLS branch WAN DCA interface
domain iwan path DCA2 direct-cloud-access
umbrella out

interface GigabitEthernet0/0/0.100  ! MPLS branch LAN interface
umbrella in direct-cloud-access default lan100
```

Additional References for Configuring Direct Cloud Access

Related Documents

Related Topic	Document Title
Performance Routing Version 3 commands	Cisco IOS Performance Routing Version 3 Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 288

Channel-based Metrics Measurement

Channel-based metrics measurement configures the performance monitors used by PfRv3 to employ a data collection method that combines the use of metadata and traffic sampled at intervals to provide traffic metrics.

Feature Information for Channel-based Metrics

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 345: Feature Information for Channel-based Metrics Measurement

Feature Name	Releases	Feature Information
Channel-based measurement of performance metrics	Cisco IOS XE Gibraltar 16.11.1	Configures the performance monitors used by PfRv3 to employ a data collection method that combines the use of metadata and traffic sampled at intervals to provide traffic metrics. New command: **channel-based-measurement**

Prerequisites for Channel-based Metrics Measurement

- Cisco IOS XE Gibraltar 16.11.1 or later

Information About Channel-based Metrics Measurement

Overview

As part of its intelligent path selection, PfRv3 uses performance monitors to gather traffic metrics. Channel-based measurement typically provides improved accuracy for metrics. The method samples packets in the traffic stream, and uses packet metadata, such as timestamp and sequence information, to generate traffic metrics. This feature uses packet-based loss measurement, not byte-loss.

Channel-based measurement of metrics provides the following benefits:

- Packets of any protocol are acceptable.
- Overcomes inaccuracies caused by methods that aggregate data from individual flows that are carried across different channels.
- Provides better tolerance of out-of-order packets.
- Reduces false threshold crossing alarms (TCAs): Previously, performance metrics have been calculated based on the samples collected in one interval. Typically, a TCA for lost packets is set for about 1% to 2%. In such a case, if there are, for example, only 30 samples in the interval and 1 packet is lost, then the packet loss rate is 3.3% and the TCA is triggered. This would be considered a false TCA because it was triggered by a single lost packet. Channel-based measurement ensures that at least 100 samples (even if these samples must be taken from different intervals) are used to calculate metrics, reducing the occurrence of false TCA.

Migration

During migration of multiple sites to a later Cisco IOS version, it may occur that the hub site and branch sites are upgraded at different times. Migrate the hub site and transit hub site first. After upgrading a hub site, if channel-based-measurement is enabled on the hub site, some branch sites might still be using IOS versions that do not support channel-based-measurement. Channel-based measurement of traffic between two branch sites requires both sites to be using Cisco IOS XE Gibraltar 16.11 or later.

How to Configure Channel-based Metrics Measurement

Channel-based Metrics Measurement Configuration

To configure the channel-based metrics measurement, use:

config terminal

domain iwan

master hub

advanced

channel-based-measurement

[**sampling-rate** *sampling-rate*] [**quick** *sampling-rate-for-quick-monitoring*]

[**sample-packet-size** *maximum-packet-size*]

Configuration Examples

Examples: Channel-based Metrics Measurement

Configure channel-based metrics measurement on a hub master controller, regardless of the number of branch sites.

Enable channel-based measurement for traffic metrics.

```
Device#config terminal
Device(config)#domain iwan
Device(config-domain)#master hub
Device(config-domain-mc)#advanced
Device(config-domain-mc-advanced)#channel-based-measurement
```

Enable channel-based measurement and configure a sampling packet size of 1300 and a sampling rate of 20 samples per second.

```
Device#config terminal
Device(config)#domain iwan
Device(config-domain)#master hub
Device(config-domain-mc)#advanced
Device(config-domain-mc-advanced)#channel-based-measurement
Device(config-domain-mc-advanced-channel-measure)#sample-packet-size 1300
Device(config-domain-mc-advanced-channel-measure)#sampling-rate 20
```

Additional References

References

Related Documents

Related Topic	Document Title
Cisco PfR commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples.	Cisco IOS Performance Routing Command Reference

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	https://www.cisco.com/c/en/us/support/index.html

CHAPTER 289

PfRv3 Event Tracing

The Event Trace for PFRv3 feature provides a trace facility for troubleshooting Performance Routing Version 3 (PfRv3). This feature enables you to monitor PfRv3 events and channels. During runtime, the event trace mechanism logs trace information in a buffer space. A display mechanism extracts and decodes the debug data.

Prerequisites for PfRv3 Event Tracing

PfRv3 event trace is enabled by default. When PfRv3 features are enabled on the route, PfRv3 writes event trace data into PfRv3's event trace buffer.

Restrictions for PfRv3 Event Tracing

By default, PfRv3 event trace can store 4096 entries of event traces. The entry size can be adjusted from 1 to 1000000 entries. Event traces are stored in memory and every event trace entry uses the memory size. The greater the number of entries, more memory is consumed. In PfRv3, each entry consumes 104 bytes. This indicates that PfRv3 event trace will consume about 416K bytes memory. Per design, the memory will have a delay allocation until first entry is written.

Information About PfRv3 Event Tracing

PfRv3 Event Tracing Options

Event Tracing uses event-trace infra framework by providing the ability to retrieve relevant part of event trace by providing show commands, such as, **show monitor event-trace pfrv3 sub-comp channel** command.

In Cisco IOS XE Fuji 16.9.1, PfRv3 supports event trace for the following subcomponents:

- process
- policy
- PDP
- channel

You can use the Event Trace for PFRv3 feature to analyze the cause of a device failure. When you configure PFRv3 features, the device records PfRv3 setup workflow and logs messages from specific subsystem components into the device memory. You can view trace messages stored in the memory by using the commands or save them to a file.

Benefits of PfRv3 Event Tracing

- Displays debug information on the console during runtime.
- Avoids multiple debug calls, and, therefore, improves device performance.
- Saves memory space.

How to Display PfRv3 Event Tracing

SUMMARY STEPS

1. **show monitor event-trace pfrv3 sub-comp channel** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]
2. **show monitor event-trace pfrv3 sub-comp pdp** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]
3. **show monitor event-trace pfrv3 sub-comp policy** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]
4. **show monitor event-trace pfrv3 sub-comp process** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]

DETAILED STEPS

Step 1 **show monitor event-trace pfrv3 sub-comp channel** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]

Example:

```
Router# show monitor event-trace pfrv3 sub-comp channel all

Jul 12 02:03:01.966: CHANNEL: INFO: BR[3] create WAN interface: name[Tunnel11] sp_color[ISP1]
ip_addr[172.16.0.1] intf_index[29] snmp index[24] CMD enabled[YES] intf_type[External] sp_tag[0x1]
zero_sla[Disable] plr[Disabled]

Jul 12 02:03:01.971: CHANNEL: INFO: BR[3] create WAN interface: name[Tunnel31] sp_color[ISP3]
ip_addr[192.168.0.1] intf_index[31] snmp index[26] CMD enabled[YES] intf_type[External] sp_tag[0x7]
 zero_sla[Disable] plr[Disabled]

Jul 12 02:03:02.469: CHANNEL: INFO: BR[3] Tunnel0 interface line protocol is coming back
```

```
Jul 12 02:03:11.685: CHANNEL: INFO: BR[2] Tunnel1 interface line protocol is coming back

Jul 12 02:03:16.272: CHANNEL: INFO: BR[2] create WAN interface: name[Tunnel10] sp_color[ISP1]
ip_addr[172.16.0.1] intf_index[28] snmp index[23] CMD enabled[YES] intf_type[External] sp_tag[0x1]
zero_sla[Disable] plr[Disabled]

Jul 12 02:03:16.282: CHANNEL: INFO: BR[2] create WAN interface: name[Tunnel30] sp_color[ISP3]
ip_addr[192.168.0.1] intf_index[30] snmp index[25] CMD enabled[YES] intf_type[External] sp_tag[0x7]
 zero_sla[Disable] plr[Disabled]

Jul 12 02:04:48.465: CHANNEL: INFO: MC[3] add channel[1]: site_id[10.30.1.1] dscp[0] intf_index[29]
 label[0x1] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:48.465: CHANNEL: INFO: MC[3] add channel[2]: site_id[10.30.1.1] dscp[0] intf_index[31]
 label[0x7] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:48.465: CHANNEL: INFO: MC[3] add channel[3]: site_id[10.30.1.1] dscp[0] intf_index[29]
 label[0x3] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:48.464: CHANNEL: INFO: MC[3] add channel[4]: site_id[10.30.1.1] dscp[0] intf_index[31]
 label[0x9] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:48.465: CHANNEL: INFO: MC[3] add channel[5]: site_id[10.30.1.1] dscp[0] intf_index[27]
 label[0x2] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:48.465: CHANNEL: INFO: MC[3] add channel[6]: site_id[10.30.1.1] dscp[0] intf_index[27]
 label[0x4] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:48.472: CHANNEL: INFO: BR[3] create channel[1] site_id[10.30.1.1] dscp[0] intf_index[29]
 label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:04:48.475: CHANNEL: INFO: BR[3] create channel[2] site_id[10.30.1.1] dscp[0] intf_index[31]
 label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:04:57.246: CHANNEL: INFO: MC[2] add channel[7]: site_id[10.30.1.1] dscp[0] intf_index[28]
 label[0x1] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:57.246: CHANNEL: INFO: MC[2] add channel[8]: site_id[10.30.1.1] dscp[0] intf_index[30]
 label[0x7] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:57.246: CHANNEL: INFO: MC[2] add channel[9]: site_id[10.30.1.1] dscp[0] intf_index[26]
 label[0x4] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:57.246: CHANNEL: INFO: MC[2] add channel[10]: site_id[10.30.1.1] dscp[0] intf_index[26]
 label[0x2] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:57.247: CHANNEL: INFO: MC[2] add channel[11]: site_id[10.30.1.1] dscp[0] intf_index[28]
 label[0x3] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:57.248: CHANNEL: INFO: MC[2] add channel[12]: site_id[10.30.1.1] dscp[0] intf_index[30]
```

```
 label[0x9] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:04:57.251: CHANNEL: INFO: BR[2] create channel[7] site_id[10.30.1.1] dscp[0] intf_index[28]
 label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:04:57.264: CHANNEL: INFO: BR[2] create channel[8] site_id[10.30.1.1] dscp[0] intf_index[30]
 label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:05:13.128: CHANNEL: INFO: MC[3] add channel[13]: site_id[10.20.1.1] dscp[0] intf_index[29]
 label[0x1] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:13.129: CHANNEL: INFO: MC[3] add channel[14]: site_id[10.20.1.1] dscp[0] intf_index[31]
 label[0x7] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:13.129: CHANNEL: INFO: MC[3] add channel[15]: site_id[10.20.1.1] dscp[0] intf_index[29]
 label[0x3] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:13.129: CHANNEL: INFO: MC[3] add channel[16]: site_id[10.20.1.1] dscp[0] intf_index[31]
 label[0x9] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:13.129: CHANNEL: INFO: MC[3] add channel[17]: site_id[10.20.1.1] dscp[0] intf_index[27]
 label[0x2] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:13.129: CHANNEL: INFO: MC[3] add channel[18]: site_id[10.20.1.1] dscp[0] intf_index[27]
 label[0x4] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:13.132: CHANNEL: INFO: BR[3] create channel[13] site_id[10.20.1.1] dscp[0] intf_index[29]
 label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:05:13.134: CHANNEL: INFO: BR[3] create channel[14] site_id[10.20.1.1] dscp[0] intf_index[31]
 label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:05:15.268: CHANNEL: INFO: MC[2] add channel[19]: site_id[10.20.1.1] dscp[0] intf_index[28]
 label[0x1] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:15.268: CHANNEL: INFO: MC[2] add channel[20]: site_id[10.20.1.1] dscp[0] intf_index[30]
 label[0x7] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:15.269: CHANNEL: INFO: MC[2] add channel[21]: site_id[10.20.1.1] dscp[0] intf_index[26]
 label[0x4] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:15.270: CHANNEL: INFO: MC[2] add channel[22]: site_id[10.20.1.1] dscp[0] intf_index[26]
 label[0x2] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:15.270: CHANNEL: INFO: MC[2] add channel[23]: site_id[10.20.1.1] dscp[0] intf_index[28]
 label[0x3] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:15.270: CHANNEL: INFO: MC[2] add channel[24]: site_id[10.20.1.1] dscp[0] intf_index[30]
```

```
 label[0x9] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[default
channel to branch] op state[Initiated and open]

Jul 12 02:05:15.272: CHANNEL: INFO: BR[2] create channel[19] site_id[10.20.1.1] dscp[0] intf_index[28]
 label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:05:15.274: CHANNEL: INFO: BR[2] create channel[20] site_id[10.20.1.1] dscp[0] intf_index[30]
 label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Initial state] TX state[Reachable]
muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:05:21.766: CHANNEL: INFO: MC[2] delete channel[23] created 00:00:06 ago: site_id[10.20.1.1]
 dscp[0] intf_index[28] label[0x3] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:05:21.766: CHANNEL: INFO: MC[2] delete channel[11] created 00:00:24 ago: site_id[10.30.1.1]
 dscp[0] intf_index[28] label[0x3] intf_type[External] channel status[Not-Available(no next-hop)] TC
 count[0] backup-TC count[0] op state[Initiated and open]

Jul 12 02:05:21.766: CHANNEL: INFO: MC[2] delete channel[12] created 00:00:24 ago: site_id[10.30.1.1]
 dscp[0] intf_index[30] label[0x9] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:05:21.767: CHANNEL: INFO: MC[2] delete channel[24] created 00:00:06 ago: site_id[10.20.1.1]
 dscp[0] intf_index[30] label[0x9] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:05:21.877: CHANNEL: INFO: MC[2] delete channel[21] created 00:00:06 ago: site_id[10.20.1.1]
 dscp[0] intf_index[26] label[0x4] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:05:21.877: CHANNEL: INFO: MC[2] delete channel[9] created 00:00:24 ago: site_id[10.30.1.1]
 dscp[0] intf_index[26] label[0x4] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:07:15.434: CHANNEL: INFO: MC[2] add channel[25]: site_id[10.30.1.1] dscp[0] intf_index[28]
 label[0x3] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[new
interface/sp-tag added] op state[Initiated and open]

Jul 12 02:07:15.434: CHANNEL: INFO: MC[2] add channel[26]: site_id[10.20.1.1] dscp[0] intf_index[28]
 label[0x3] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[new
interface/sp-tag added] op state[Initiated and open]

Jul 12 02:07:15.438: CHANNEL: INFO: MC[2] add channel[27]: site_id[10.30.1.1] dscp[0] intf_index[30]
 label[0x9] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[new
interface/sp-tag added] op state[Initiated and open]

Jul 12 02:07:15.438: CHANNEL: INFO: MC[2] add channel[28]: site_id[10.20.1.1] dscp[0] intf_index[30]
 label[0x9] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[new
interface/sp-tag added] op state[Initiated and open]

Jul 12 02:07:15.788: CHANNEL: INFO: MC[2] add channel[29]: site_id[10.30.1.1] dscp[0] intf_index[26]
 label[0x4] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[new
interface/sp-tag added] op state[Initiated and open]

Jul 12 02:07:15.788: CHANNEL: INFO: MC[2] add channel[30]: site_id[10.20.1.1] dscp[0] intf_index[26]
 label[0x4] intf_type[External] IDC channel[NO] MHOP channel[NO] To-HUB channel[NO] reason[new
interface/sp-tag added] op state[Initiated and open]

Jul 12 02:08:47.133: CHANNEL: INFO: BR[2] delete channel[7] when clear BR DB: site_id[10.30.1.1]
dscp[0] intf_index[28] label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Reachable] TX
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:08:47.135: CHANNEL: INFO: BR[2] delete channel[8] when clear BR DB: site_id[10.30.1.1]
```

```
dscp[0] intf_index[30] label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Reachable] TX
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:08:47.136: CHANNEL: INFO: BR[2] delete channel[19] when clear BR DB: site_id[10.20.1.1]
dscp[0] intf_index[28] label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Reachable] TX
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:08:47.137: CHANNEL: INFO: BR[2] delete channel[20] when clear BR DB: site_id[10.20.1.1]
dscp[0] intf_index[30] label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Reachable] TX
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]

Jul 12 02:08:47.174: CHANNEL: INFO: MC[2] delete channel[30] created 00:01:31 ago: site_id[10.20.1.1]
 dscp[0] intf_index[26] label[0x4] intf_type[External] channel status[Not-Available(no next-hop)] TC
 count[0] backup-TC count[0] op state[Initiated and open]

Jul 12 02:08:47.174: CHANNEL: INFO: MC[2] delete channel[29] created 00:01:31 ago: site_id[10.30.1.1]
 dscp[0] intf_index[26] label[0x4] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:08:47.175: CHANNEL: INFO: MC[2] delete channel[19] created 00:03:31 ago: site_id[10.20.1.1]
 dscp[0] intf_index[28] label[0x1] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:08:47.175: CHANNEL: INFO: MC[2] delete channel[7] created 00:03:49 ago: site_id[10.30.1.1]
 dscp[0] intf_index[28] label[0x1] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:08:47.175: CHANNEL: INFO: MC[2] delete channel[8] created 00:03:49 ago: site_id[10.30.1.1]
 dscp[0] intf_index[30] label[0x7] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:08:47.175: CHANNEL: INFO: MC[2] delete channel[20] created 00:03:31 ago: site_id[10.20.1.1]
 dscp[0] intf_index[30] label[0x7] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]

Jul 12 02:08:47.175: CHANNEL: INFO: MC[2] delete channel[22] created 00:03:31 ago: site_id[10.20.1.1]
 dscp[0] intf_index[26] label[0x2] intf_type[External] channel status[Not-Available(no next-hop)] TC
 count[0] backup-TC count[0] op state[Initiated and open]

Jul 12 02:08:47.175: CHANNEL: INFO: MC[2] delete channel[10] created 00:03:49 ago: site_id[10.30.1.1]
 dscp[0] intf_index[26] label[0x2] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:47.176: CHANNEL: INFO: MC[2] delete channel[26] created 00:01:31 ago: site_id[10.20.1.1]
 dscp[0] intf_index[28] label[0x3] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:47.176: CHANNEL: INFO: MC[2] delete channel[25] created 00:01:31 ago: site_id[10.30.1.1]
 dscp[0] intf_index[28] label[0x3] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:47.177: CHANNEL: INFO: MC[2] delete channel[27] created 00:01:31 ago: site_id[10.30.1.1]
 dscp[0] intf_index[30] label[0x9] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:47.177: CHANNEL: INFO: MC[2] delete channel[28] created 00:01:31 ago: site_id[10.20.1.1]
 dscp[0] intf_index[30] label[0x9] intf_type[External] channel status[Not-Available(no next-hop)] TC
 count[0] backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:47.211: CHANNEL: INFO: BR[2] Tunnel1 interface line protocol is going down, may enqueue
 ALL_CHAN_UNREACH msg
Jul 12 02:08:48.235: CHANNEL: INFO: BR[3] delete channel[1] when clear BR DB: site_id[10.30.1.1]
dscp[0] intf_index[29] label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Reachable] TX
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]
Jul 12 02:08:48.235: CHANNEL: INFO: BR[3] delete channel[2] when clear BR DB: site_id[10.30.1.1]
dscp[0] intf_index[31] label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Reachable] TX
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]
Jul 12 02:08:48.235: CHANNEL: INFO: BR[3] delete channel[13] when clear BR DB: site_id[10.20.1.1]
dscp[0] intf_index[29] label[0x1] sp_color[ISP1] next-hop[172.16.0.1] RX state[Reachable] TX
```

```
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]
Jul 12 02:08:48.235: CHANNEL: INFO: BR[3] delete channel[14] when clear BR DB: site_id[10.20.1.1]
dscp[0] intf_index[31] label[0x7] sp_color[ISP3] next-hop[192.168.0.1] RX state[Reachable] TX
state[Reachable] muted-by-0-sla[NO] op state[Initiated and open]
Jul 12 02:08:48.259: CHANNEL: INFO: MC[3] delete channel[6] created 00:03:59 ago: site_id[10.30.1.1]
 dscp[0] intf_index[27] label[0x4] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.259: CHANNEL: INFO: MC[3] delete channel[18] created 00:03:35 ago: site_id[10.20.1.1]
 dscp[0] intf_index[27] label[0x4] intf_type[External] channel status[Not-Available(no next-hop)] TC
 count[0] backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.260: CHANNEL: INFO: MC[3] delete channel[1] created 00:03:59 ago: site_id[10.30.1.1]
 dscp[0] intf_index[29] label[0x1] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.260: CHANNEL: INFO: MC[3] delete channel[13] created 00:03:35 ago: site_id[10.20.1.1]
 dscp[0] intf_index[29] label[0x1] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[2] created 00:03:59 ago: site_id[10.30.1.1]
 dscp[0] intf_index[31] label[0x7] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[14] created 00:03:35 ago: site_id[10.20.1.1]
 dscp[0] intf_index[31] label[0x7] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[5] created 00:03:59 ago: site_id[10.30.1.1]
 dscp[0] intf_index[27] label[0x2] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[17] created 00:03:35 ago: site_id[10.20.1.1]
 dscp[0] intf_index[27] label[0x2] intf_type[External] channel status[Not-Available(no next-hop)] TC
 count[0] backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[3] created 00:03:59 ago: site_id[10.30.1.1]
 dscp[0] intf_index[29] label[0x3] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[15] created 00:03:35 ago: site_id[10.20.1.1]
 dscp[0] intf_index[29] label[0x3] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[4] created 00:03:59 ago: site_id[10.30.1.1]
 dscp[0] intf_index[31] label[0x9] intf_type[External] channel status[Not-Available] TC count[0]
backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.261: CHANNEL: INFO: MC[3] delete channel[16] created 00:03:35 ago: site_id[10.20.1.1]
 dscp[0] intf_index[31] label[0x9] intf_type[External] channel status[Not-Available(no next-hop)] TC
 count[0] backup-TC count[0] op state[Initiated and open]
Jul 12 02:08:48.288: CHANNEL: INFO: BR[3] Tunnel0 interface line protocol is going down, may enqueue
 ALL_CHAN_UNREACH msg
```

Displays event trace for PfRv3 channels.

Step 2 **show monitor event-trace pfrv3 sub-comp pdp** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]

Example:

```
Router# show monitor event-trace pfrv3 sub-comp pdp all

Jul 12 02:23:01.024: PDP: INFO: MC[6] TC[2] PDP RESULT: state is un-controlled, PDP trigger reason
is New TC Learned, violate type is None
Jul 12 02:23:31.948: PDP: INFO: MC[6] TC[2]: +Channel[55]: Path-Pref[Primary], Usable[P], Reachable[P],
 TCA Loss[-], TCA Delay[-], TCA Jitter[-], 95Bandwidth[P], Prefix Reachable[P]
Jul 12 02:23:31.948: PDP: INFO: MC[6] TC[2]: -Channel[57]: Path-Pref[Fallback], Usable[P], Reachable[P],
 TCA Loss[-], TCA Delay[-], TCA Jitter[-], 95Bandwidth[P], Prefix Reachable[P]
Jul 12 02:23:31.948: PDP: INFO: MC[6] TC[2]: -Channel[58]: Path-Pref[Fallback], Usable[P], Reachable[P],
 TCA Loss[-], TCA Delay[-], TCA Jitter[-], 95Bandwidth[P], Prefix Reachable[P]
Jul 12 02:23:31.948: PDP: INFO: MC[6] TC[2]: *Channel[59]: Path-Pref[Primary], Usable[P], Reachable[P],
 TCA Loss[-], TCA Delay[-], TCA Jitter[-], 95Bandwidth[P], Prefix Reachable[P]
Jul 12 02:23:31.948: PDP: INFO: MC[6] TC[2] PDP RESULT: state is controlled, PDP trigger reason is
Backoff Timer Expired, violate type is Uncontrolled to Controlled Transition
```

Displays event trace for PfRv3 policy decision points (PDP).

Step 3 **show monitor event-trace pfrv3 sub-comp policy** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]

Example:

```
Router# show monitor event-trace pfrv3 sub-comp policy all

Jul 12 02:02:42.727: POLICY: INFO: MC[2]: Pol_map seq 10: Set sp pref to ISP1 at index 0
Jul 12 02:02:42.727: POLICY: INFO: MC[2]: Pol_map seq 10: Set sp fallback to ISP2 at index 0
Jul 12 02:02:42.890: POLICY: INFO: MC[2]: Pol_map seq 20: Set sp pref to ISP1 at index 0
Jul 12 02:03:01.959: POLICY: INFO: BR[3]: Create C3PL policy for Egress direction
Jul 12 02:03:01.959: POLICY: INFO: BR[3]: Create C3PL policy for Ingress direction
Jul 12 02:03:01.967: POLICY: INFO: BR[3]: Create C3PL policy for Egress direction
Jul 12 02:03:01.967: POLICY: INFO: BR[3]: Create C3PL policy for Ingress direction
Jul 12 02:03:08.212: POLICY: INFO: MC[2]: MC policy downloaded:site id[10.10.1.1], domain[default],
 vrf[green]
Jul 12 02:03:08.212: POLICY: INFO: MC[2]: Policy publish max allowed xml size[3030], exact xml
size[2099]
Jul 12 02:03:09.355: POLICY: INFO: MC[3]: MC policy downloaded:site id[10.10.1.1], domain[default],
 vrf[red]
Jul 12 02:03:09.355: POLICY: INFO: MC[3]: Policy publish max allowed xml size[2303], exact xml
size[1571]
Jul 12 02:03:10.927: POLICY: INFO: BR[2]: Updating PMI policies

Jul 12 02:03:10.927: POLICY: INFO: BR[2]: Create C3PL policy for Ingress direction

Jul 12 02:03:10.949: POLICY: INFO: BR[2]: Create flow monitor MON-Ingress-per-DSCP-2-48-0

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create filter dscp:46,appid:0

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create class CENT-Class-Ingress-DSCP-ef-2-2

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create filter dscp:40,appid:0

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create class CENT-Class-Ingress-DSCP-cs5-2-3

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create filter dscp:32,appid:0

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create class CENT-Class-Ingress-DSCP-cs4-2-4

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create react for class CENT-Class-Ingress-DSCP-ef-2-2

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react packet-loss-rate val=100 id=2

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react one-way-delay val=20 id=3

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react network-delay-avg val=40 id=4

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react jitter val=5000 id=5

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react byte-loss-rate val=100 id=6

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create react for class CENT-Class-Ingress-DSCP-cs5-2-3

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react jitter val=5000 id=7

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create react for class CENT-Class-Ingress-DSCP-cs4-2-4

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react one-way-delay val=20 id=8

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Provision react network-delay-avg val=40 id=9
```

```
Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create PMI policy CENT-Policy-Ingress-2-2

Jul 12 02:03:10.950: POLICY: INFO: BR[2]: Create C3PL policy for Egress direction

Jul 12 02:03:10.976: POLICY: INFO: BR[2]: Create flow monitor MON-Egress-aggregate-2-48-1

Jul 12 02:03:10.976: POLICY: INFO: BR[2]: Create filter dscp:0,appid:0

Jul 12 02:03:10.976: POLICY: INFO: BR[2]: Create class CENT-Class-Egress-ANY-2-5

Jul 12 02:03:10.977: POLICY: INFO: BR[2]: Create PMI policy CENT-Policy-Egress-2-3

Jul 12 02:03:12.013: POLICY: INFO: BR[3]: Updating PMI policies

Jul 12 02:03:12.013: POLICY: INFO: BR[3]: Create C3PL policy for Ingress direction

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Create flow monitor MON-Ingress-per-DSCP-3-48-2

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Create filter dscp:46,appid:0

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Create class CENT-Class-Ingress-DSCP-ef-3-6

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Create filter dscp:40,appid:0

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Create class CENT-Class-Ingress-DSCP-cs5-3-7

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Create react for class CENT-Class-Ingress-DSCP-ef-3-6

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Provision react packet-loss-rate val=100 id=10

Jul 12 02:03:12.014: POLICY: INFO: BR[3]: Provision react byte-loss-rate val=100 id=11

Jul 12 02:03:12.015: POLICY: INFO: BR[3]: Create react for class CENT-Class-Ingress-DSCP-cs5-3-7

Jul 12 02:03:12.015: POLICY: INFO: BR[3]: Provision react packet-loss-rate val=100 id=12
Jul 12 02:03:12.015: POLICY: INFO: BR[3]: Provision react byte-loss-rate val=100 id=13
Jul 12 02:03:12.015: POLICY: INFO: BR[3]: Create PMI policy CENT-Policy-Ingress-3-4
Jul 12 02:03:12.094: POLICY: INFO: BR[3]: Activate flow monitor MON-Ingress-per-DSCP-3-48-2
Jul 12 02:03:12.094: POLICY: INFO: BR[3]: Activate PMI policy CENT-Policy-Ingress-3-4 on Tunnel31
Jul 12 02:03:12.094: POLICY: INFO: BR[3]: Activate PMI policy CENT-Policy-Ingress-3-4 on Tunnel11
Jul 12 02:03:12.094: POLICY: INFO: BR[3]: Create C3PL policy for Egress direction
Jul 12 02:03:12.119: POLICY: INFO: BR[3]: Create flow monitor MON-Egress-aggregate-3-48-3
Jul 12 02:03:12.118: POLICY: INFO: BR[3]: Create filter dscp:0,appid:0
Jul 12 02:03:12.118: POLICY: INFO: BR[3]: Create class CENT-Class-Egress-ANY-3-8
Jul 12 02:03:12.118: POLICY: INFO: BR[3]: Create PMI policy CENT-Policy-Egress-3-5
Jul 12 02:03:12.151: POLICY: INFO: BR[3]: Activate PMI policy CENT-Policy-Egress-3-5 on Tunnel31
Jul 12 02:03:12.151: POLICY: INFO: BR[3]: Activate PMI policy CENT-Policy-Egress-3-5 on Tunnel11
Jul 12 02:03:16.251: POLICY: INFO: BR[2]: Activate PMI policy CENT-Policy-Egress-2-3 on Tunnel10
Jul 12 02:03:16.265: POLICY: INFO: BR[2]: Activate flow monitor MON-Ingress-per-DSCP-2-48-0
Jul 12 02:03:16.265: POLICY: INFO: BR[2]: Activate PMI policy CENT-Policy-Ingress-2-2 on Tunnel10
Jul 12 02:03:16.274: POLICY: INFO: BR[2]: Activate PMI policy CENT-Policy-Egress-2-3 on Tunnel30
Jul 12 02:03:16.275: POLICY: INFO: BR[2]: Activate PMI policy CENT-Policy-Ingress-2-2 on Tunnel30
Jul 12 02:08:47.180: POLICY: INFO: BR[2]: De-activate PMI policy CENT-Policy-Egress-2-3 on Tunnel30
Jul 12 02:08:47.182: POLICY: INFO: BR[2]: De-activate PMI policy CENT-Policy-Ingress-2-2 on Tunnel30
Jul 12 02:08:47.188: POLICY: INFO: BR[2]: Delete class CENT-Class-Egress-ANY-2-5
Jul 12 02:08:47.190: POLICY: INFO: BR[2]: Delete flow monitor MON-Egress-aggregate-2-48-1
Jul 12 02:08:47.190: POLICY: INFO: BR[2]: De-activate PMI policy CENT-Policy-Egress-2-3 on Tunnel10
Jul 12 02:08:47.190: POLICY: INFO: BR[2]: De-activate flow monitor MON-Ingress-per-DSCP-2-48-0
Jul 12 02:08:47.201: POLICY: INFO: BR[2]: Delete react from class CENT-Class-Ingress-DSCP-ef-2-2
Jul 12 02:08:47.201: POLICY: INFO: BR[2]: Delete class CENT-Class-Ingress-DSCP-ef-2-2
Jul 12 02:08:47.201: POLICY: INFO: BR[2]: Delete react from class CENT-Class-Ingress-DSCP-cs5-2-3
Jul 12 02:08:47.201: POLICY: INFO: BR[2]: Delete class CENT-Class-Ingress-DSCP-cs5-2-3
Jul 12 02:08:47.201: POLICY: INFO: BR[2]: Delete react from class CENT-Class-Ingress-DSCP-cs4-2-4
Jul 12 02:08:47.201: POLICY: INFO: BR[2]: Delete class CENT-Class-Ingress-DSCP-cs4-2-4
```

```
Jul 12 02:08:47.201: POLICY: INFO: BR[2]: Delete flow monitor MON-Ingress-per-DSCP-2-48-0
Jul 12 02:08:47.202: POLICY: INFO: BR[2]: De-activate PMI policy CENT-Policy-Ingress-2-2 on Tunnel10
Jul 12 02:08:48.264: POLICY: INFO: BR[3]: De-activate PMI policy CENT-Policy-Egress-3-5 on Tunnel31
Jul 12 02:08:48.265: POLICY: INFO: BR[3]: De-activate PMI policy CENT-Policy-Ingress-3-4 on Tunnel31
Jul 12 02:08:48.281: POLICY: INFO: BR[3]: Delete class CENT-Class-Egress-ANY-3-8
Jul 12 02:08:48.281: POLICY: INFO: BR[3]: Delete flow monitor MON-Egress-aggregate-3-48-3
Jul 12 02:08:48.281: POLICY: INFO: BR[3]: De-activate PMI policy CENT-Policy-Egress-3-5 on Tunnel11
Jul 12 02:08:48.281: POLICY: INFO: BR[3]: De-activate flow monitor MON-Ingress-per-DSCP-3-48-2
Jul 12 02:08:48.287: POLICY: INFO: BR[3]: Delete react from class CENT-Class-Ingress-DSCP-ef-3-6
Jul 12 02:08:48.287: POLICY: INFO: BR[3]: Delete class CENT-Class-Ingress-DSCP-ef-3-6
Jul 12 02:08:48.287: POLICY: INFO: BR[3]: Delete react from class CENT-Class-Ingress-DSCP-cs5-3-7
Jul 12 02:08:48.287: POLICY: INFO: BR[3]: Delete class CENT-Class-Ingress-DSCP-cs5-3-7
Jul 12 02:08:48.287: POLICY: INFO: BR[3]: Delete flow monitor MON-Ingress-per-DSCP-3-48-2
Jul 12 02:08:48.286: POLICY: INFO: BR[3]: De-activate PMI policy CENT-Policy-Ingress-3-4 on Tunnel11
```

Displays event trace for PfRv3 policies.

Step 4 **show monitor event-trace pfrv3 sub-comp process** {**all** | **back** *duration* | **clock** *duration* | **from-boot** *seconds* | **latest**} [**detail**]

Example:

```
Router# show monitor event-trace pfrv3 sub-comp process all

Jul 12 02:02:42.467: PROCESS: INFO: BR[2] Register CMD client : client id 4, result Succeed

Jul 12 02:02:42.473: PROCESS: INFO: BR[2] started

Jul 12 02:02:42.965: PROCESS: INFO: BR[3] started

Jul 12 02:02:57.957: PROCESS: INFO: MC[2] Eigrp autocfg opcode: Listening, interface: Loopback1,
split: FALSE, action: Add, result: Succeed
Jul 12 02:02:57.957: PROCESS: INFO: MC[2] Eigrp autocfg opcode: Stub, interface: Loopback1, split:
FALSE, stub stubbed: FALSE, stub connected: FALSE, stub leaking: FALSE, action: Delete


Jul 12 02:02:57.957: PROCESS: INFO: MC[2] SAF peering succeed listener: 10.10.1.1

Jul 12 02:02:57.958: PROCESS: INFO: MC[2] SAF peering subscribe sub-service: cent-policy succeed

Jul 12 02:02:57.968: PROCESS: INFO: MC[2] SAF peering publish sub-service: globals, origin: 10.10.1.1,
 size: 997, compressed size: 417, publish seq: 1, publish reason: Normal, result: Peering Success


Jul 12 02:02:57.968: PROCESS: INFO: MC[2] SAF peering subscribe sub-service: site-prefix succeed

Jul 12 02:02:57.968: PROCESS: INFO: MC[2] SAF peering subscribe sub-service: Capability succeed

Jul 12 02:02:57.968: PROCESS: INFO: MC[2] SAF peering subscribe sub-service: globals succeed

Jul 12 02:02:57.970: PROCESS: INFO: MC server listening on: 10.10.1.1, port: 17749

Jul 12 02:02:57.970: PROCESS: INFO: MC[2] SAF peering publish sub-service: globals, origin: 10.10.1.1,
 size: 997, compressed size: 417, publish seq: 2, publish reason: Normal, result: Peering Success


Jul 12 02:02:57.970: PROCESS: INFO: MC[2] SAF peering publish sub-service: globals, origin: 10.10.1.1,
 size: 997, compressed size: 417, publish seq: 3, publish reason: Normal, result: Peering Success


Jul 12 02:02:58.490: PROCESS: INFO: MC[3] Eigrp autocfg opcode: Listening, interface: Loopback2,
split: FALSE, action: Add, result: Succeed
Jul 12 02:02:58.492: PROCESS: INFO: MC[3] Eigrp autocfg opcode: Stub, interface: Loopback2, split:
FALSE, stub stubbed: FALSE, stub connected: FALSE, stub leaking: FALSE, action: Delete
```

```
Jul 12 02:02:58.492: PROCESS: INFO: MC[3] SAF peering succeed listener: 10.10.1.1

Jul 12 02:02:58.492: PROCESS: INFO: MC[3] SAF peering subscribe sub-service: cent-policy succeed

Jul 12 02:02:58.500: PROCESS: INFO: MC[3] SAF peering publish sub-service: globals, origin: 10.10.1.1,
 size: 996, compressed size: 419, publish seq: 1, publish reason: Normal, result: Peering Success


Jul 12 02:02:58.500: PROCESS: INFO: MC[3] SAF peering subscribe sub-service: site-prefix succeed

Jul 12 02:02:58.500: PROCESS: INFO: MC[3] SAF peering subscribe sub-service: Capability succeed

Jul 12 02:02:58.500: PROCESS: INFO: MC[3] SAF peering subscribe sub-service: globals succeed

Jul 12 02:02:58.501: PROCESS: INFO: MC server listening on: 10.10.1.1, port: 17749

Jul 12 02:03:00.020: PROCESS: INFO: MC[2] SAF peering subscribe sub-service: pmi succeed

Jul 12 02:03:01.455: PROCESS: INFO: BR[3] no shutdown

Jul 12 02:03:01.462: PROCESS: INFO: BR[3] start communication process

Jul 12 02:03:01.463: PROCESS: INFO: BR[3] start exporter process

Jul 12 02:03:01.535: PROCESS: INFO: BR[3] start route learn

Jul 12 02:03:01.540: PROCESS: INFO: MC[3] SAF peering subscribe sub-service: pmi succeed

Jul 12 02:03:01.738: PROCESS: INFO: BR[3] ip: 10.10.1.1, port: 20660, connected to MC ip: 10.10.1.1,
 port: 17749
Jul 12 02:03:01.739: PROCESS: INFO: BR[3] Eigrp autocfg opcode: Stub, interface: Loopback2, split:
FALSE, stub stubbed: FALSE, stub connected: FALSE, stub leaking: FALSE, action: Delete


Jul 12 02:03:01.739: PROCESS: INFO: BR[3] Eigrp autocfg opcode: Neighbor multihop, Hub ip: 10.10.1.1,
 interface: Loopback2, split: FALSE, action: Add, result: Succeed


Jul 12 02:03:01.739: PROCESS: INFO: BR[3] SAF peering succeed: 10.10.1.1 to 10.10.1.1

Jul 12 02:03:01.740: PROCESS: INFO: BR[3] SAF peering subscribe sub-service: pmi succeed

Jul 12 02:03:01.740: PROCESS: INFO: BR[3] SAF peering subscribe sub-service: site-prefix succeed

Jul 12 02:03:01.740: PROCESS: INFO: BR[3] SAF peering received sub-service: globals, from: 10.10.1.1,
 data size: 439, data seq: 1
Jul 12 02:03:01.741: PROCESS: INFO: BR[3] SAF peering subscribe sub-service: globals succeed

Jul 12 02:03:01.741: PROCESS: INFO: BR[3] SAF peering subscribe sub-service: Capability succeed

Jul 12 02:03:01.958: PROCESS: INFO: BR[3] Register CMD IDB : idb Tunnel0, client id 4, result Succeed

Jul 12 02:03:01.966: PROCESS: INFO: BR[3] Register CMD IDB : idb Tunnel11, client id 4, result Succeed

Jul 12 02:03:01.971: PROCESS: INFO: BR[3] Register CMD IDB : idb Tunnel31, client id 4, result Succeed

Jul 12 02:03:04.116: PROCESS: INFO: MC[2] SAF peering publish sub-service: site-prefix, origin:
10.10.1.1, size: 333, compressed size: 174, publish seq: 4, publish reason: On-Demand, result: Peering
 Success

Jul 12 02:03:04.117: PROCESS: INFO: MC[2] SAF peering publish sub-service: Capability, origin:
10.10.1.1, size: 274, compressed size: 159, publish seq: 5, publish reason: Periodic, result: Peering
```

```
 Success

Jul 12 02:03:05.247: PROCESS: INFO: BR[3] SAF peering received sub-service: site-prefix, from:
10.10.1.1, data size: 190, data seq: 2
Jul 12 02:03:05.256: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.10.1.1/32 from
10.10.1.1
Jul 12 02:03:05.257: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.10.1.1
Jul 12 02:03:05.258: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.10.1.1
Jul 12 02:03:05.258: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.10.1.1
Jul 12 02:03:05.259: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.10.1.1
Jul 12 02:03:05.259: PROCESS: INFO: MC[3] SAF peering publish sub-service: site-prefix, origin:
10.10.1.1, size: 333, compressed size: 170, publish seq: 2, publish reason: On-Demand, result: Peering
 Success

Jul 12 02:03:05.258: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.10.1.1,
 data size: 260, data seq: 3
Jul 12 02:03:05.259: PROCESS: INFO: MC[3] SAF peering publish sub-service: Capability, origin:
10.10.1.1, size: 618, compressed size: 240, publish seq: 3, publish reason: Periodic, result: Peering
 Success

Jul 12 02:03:08.213: PROCESS: INFO: MC[2] SAF peering publish sub-service: cent-policy, origin:
10.10.1.1, size: 210, compressed size: 455, publish seq: 6, publish reason: Normal, result: Peering
 Success

Jul 12 02:03:09.355: PROCESS: INFO: MC[3] SAF peering publish sub-service: cent-policy, origin:
10.10.1.1, size: 1572, compressed size: 372, publish seq: 4, publish reason: Normal, result: Peering
 Success

Jul 12 02:03:09.610: PROCESS: INFO: MC[3] SAF peering received sub-service: site-prefix, from:
10.15.1.1, data size: 193, data seq: 1
Jul 12 02:03:09.610: PROCESS: INFO: BR[3] SAF peering received sub-service: site-prefix, from:
10.15.1.1, data size: 193, data seq: 1
Jul 12 02:03:09.610: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:09.610: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:09.610: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:09.610: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
Jul 12 02:03:09.610: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:09.611: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:09.611: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:09.611: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:09.611: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
Jul 12 02:03:09.611: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:09.614: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:09.614: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:09.614: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:09.614: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
```

```
Jul 12 02:03:09.615: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:09.615: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:09.615: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:09.615: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:09.615: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
Jul 12 02:03:09.615: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:09.626: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 251, data seq: 2
Jul 12 02:03:09.626: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 251, data seq: 2
Jul 12 02:03:10.180: PROCESS: INFO: BR[2] no shutdown

Jul 12 02:03:10.185: PROCESS: INFO: BR[2] start communication process

Jul 12 02:03:10.185: PROCESS: INFO: BR[2] start exporter process

Jul 12 02:03:10.262: PROCESS: INFO: MC[2] SAF peering publish sub-service: pmi, origin: 10.10.1.1,
size: 1874, compressed size: 494, publish seq: 7, publish reason: Normal, result: Peering Success


Jul 12 02:03:10.724: PROCESS: INFO: BR[2] start route learn

Jul 12 02:03:10.925: PROCESS: INFO: BR[2] ip: 10.10.1.1, port: 52402, connected to MC ip: 10.10.1.1,
 port: 17749
Jul 12 02:03:10.925: PROCESS: INFO: BR[2] Eigrp autocfg opcode: Stub, interface: Loopback1, split:
FALSE, stub stubbed: FALSE, stub connected: FALSE, stub leaking: FALSE, action: Delete


Jul 12 02:03:10.924: PROCESS: INFO: BR[2] Eigrp autocfg opcode: Neighbor multihop, Hub ip: 10.10.1.1,
 interface: Loopback1, split: FALSE, action: Add, result: Succeed


Jul 12 02:03:10.924: PROCESS: INFO: BR[2] SAF peering succeed: 10.10.1.1 to 10.10.1.1

Jul 12 02:03:10.925: PROCESS: INFO: BR[2] SAF peering received sub-service: pmi, from: 10.10.1.1,
data size: 514, data seq: 7
Jul 12 02:03:10.925: PROCESS: INFO: BR[2] SAF peering subscribe sub-service: pmi succeed

Jul 12 02:03:10.925: PROCESS: INFO: BR[2] SAF peering received sub-service: site-prefix, from:
10.10.1.1, data size: 194, data seq: 4
Jul 12 02:03:10.925: PROCESS: INFO: BR[2] SAF peering subscribe sub-service: site-prefix succeed

Jul 12 02:03:10.925: PROCESS: INFO: BR[2] SAF peering received sub-service: globals, from: 10.10.1.1,
 data size: 437, data seq: 3
Jul 12 02:03:10.926: PROCESS: INFO: BR[2] SAF peering subscribe sub-service: globals succeed

Jul 12 02:03:10.926: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.10.1.1,
 data size: 179, data seq: 5
Jul 12 02:03:10.926: PROCESS: INFO: BR[2] SAF peering subscribe sub-service: Capability succeed

Jul 12 02:03:10.977: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.10.1.1/32 from
10.10.1.1
Jul 12 02:03:10.978: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.10.1.1
Jul 12 02:03:10.978: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.10.1.1
Jul 12 02:03:10.978: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.10.1.1
```

```
Jul 12 02:03:10.979: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.10.1.1
Jul 12 02:03:12.011: PROCESS: INFO: BR[3] SAF peering received sub-service: pmi, from: 10.10.1.1,
data size: 460, data seq: 5
Jul 12 02:03:12.131: PROCESS: INFO: MC[3] SAF peering publish sub-service: pmi, origin: 10.10.1.1,
size: 1682, compressed size: 440, publish seq: 5, publish reason: Normal, result: Peering Success


Jul 12 02:03:16.244: PROCESS: INFO: BR[2] Register CMD IDB : idb Tunnel1, client id 4, result Succeed

Jul 12 02:03:16.272: PROCESS: INFO: BR[2] Register CMD IDB : idb Tunnel10, client id 4, result Succeed

Jul 12 02:03:16.281: PROCESS: INFO: BR[2] Register CMD IDB : idb Tunnel30, client id 4, result Succeed

Jul 12 02:03:24.577: PROCESS: INFO: MC[2] SAF peering received sub-service: site-prefix, from:
10.15.1.1, data size: 193, data seq: 1
Jul 12 02:03:24.577: PROCESS: INFO: BR[2] SAF peering received sub-service: site-prefix, from:
10.15.1.1, data size: 193, data seq: 1
Jul 12 02:03:24.577: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:24.577: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:24.577: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:24.578: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
Jul 12 02:03:24.578: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:24.579: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:24.579: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:24.579: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:24.579: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
Jul 12 02:03:24.579: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:24.581: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:24.581: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:24.581: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:24.581: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
Jul 12 02:03:24.581: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:24.582: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.15.1.1/32 from
10.15.1.1
Jul 12 02:03:24.582: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.10.0.0/16 from
10.15.1.1
Jul 12 02:03:24.582: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.15.0.0/16 from
10.15.1.1
Jul 12 02:03:24.582: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.16.0.0/16 from
10.15.1.1
Jul 12 02:03:24.582: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.0.0.0/8 from
10.15.1.1
Jul 12 02:03:24.582: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 250, data seq: 2
Jul 12 02:03:24.582: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 250, data seq: 2
Jul 12 02:03:40.913: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 280, data seq: 3
```

```
Jul 12 02:03:40.913: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 280, data seq: 3
Jul 12 02:03:51.318: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.10.1.1,
 data size: 290, data seq: 6
Jul 12 02:03:51.319: PROCESS: INFO: MC[3] SAF peering publish sub-service: Capability, origin:
10.10.1.1, size: 822, compressed size: 270, publish seq: 6, publish reason: On-Demand, result: Peering
 Success

Jul 12 02:03:55.763: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 278, data seq: 3
Jul 12 02:03:55.763: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.15.1.1,
 data size: 278, data seq: 3
Jul 12 02:04:00.542: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.10.1.1,
 data size: 287, data seq: 8
Jul 12 02:04:00.543: PROCESS: INFO: MC[2] SAF peering publish sub-service: Capability, origin:
10.10.1.1, size: 822, compressed size: 267, publish seq: 8, publish reason: On-Demand, result: Peering
 Success

Jul 12 02:04:48.460: PROCESS: INFO: MC[3] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 159, data seq: 1
Jul 12 02:04:48.460: PROCESS: INFO: BR[3] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 159, data seq: 1
Jul 12 02:04:48.461: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:48.461: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:48.462: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:48.463: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:48.463: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 183, data seq: 2
Jul 12 02:04:48.464: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 183, data seq: 2
Jul 12 02:04:57.245: PROCESS: INFO: MC[2] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 159, data seq: 1
Jul 12 02:04:57.245: PROCESS: INFO: BR[2] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 159, data seq: 1
Jul 12 02:04:57.245: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:57.245: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:57.246: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:57.247: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:04:57.247: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 183, data seq: 2
Jul 12 02:04:57.246: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 183, data seq: 2
Jul 12 02:05:00.536: PROCESS: INFO: MC[3] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 184, data seq: 3
Jul 12 02:05:00.536: PROCESS: INFO: BR[3] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 184, data seq: 3
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.0.1/32 from
10.30.1.1
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.0.0/24 from
10.30.1.1
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.2.1/32 from
10.30.1.1
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
```

```
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.0.1/32 from
10.30.1.1
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.0.0/24 from
10.30.1.1
Jul 12 02:05:00.537: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.30.2.1/32 from
10.30.1.1
Jul 12 02:05:00.537: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:05:00.538: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.0.1/32 from
10.30.1.1
Jul 12 02:05:00.539: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.0.0/24 from
10.30.1.1
Jul 12 02:05:00.540: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.2.1/32 from
10.30.1.1
Jul 12 02:05:00.540: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:05:00.540: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.0.1/32 from
10.30.1.1
Jul 12 02:05:00.540: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.0.0/24 from
10.30.1.1
Jul 12 02:05:00.540: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.30.2.1/32 from
10.30.1.1
Jul 12 02:05:00.541: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 306, data seq: 4
Jul 12 02:05:00.541: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 306, data seq: 4
Jul 12 02:05:11.981: PROCESS: INFO: MC[2] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 171, data seq: 3
Jul 12 02:05:11.981: PROCESS: INFO: BR[2] SAF peering received sub-service: site-prefix, from:
10.30.1.1, data size: 171, data seq: 3
Jul 12 02:05:11.982: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:05:11.982: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.30.0.0/16 from
10.30.1.1
Jul 12 02:05:11.982: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:05:11.983: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.30.0.0/16 from
10.30.1.1
Jul 12 02:05:11.983: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:05:11.985: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.30.0.0/16 from
10.30.1.1
Jul 12 02:05:11.985: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.30.1.1/32 from
10.30.1.1
Jul 12 02:05:11.985: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.30.0.0/16 from
10.30.1.1
Jul 12 02:05:11.985: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 297, data seq: 4
Jul 12 02:05:11.985: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 297, data seq: 4
Jul 12 02:05:13.127: PROCESS: INFO: MC[3] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 159, data seq: 1
Jul 12 02:05:13.127: PROCESS: INFO: BR[3] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 159, data seq: 1
Jul 12 02:05:13.126: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:13.126: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:13.128: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:13.128: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:13.128: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 184, data seq: 2
```

```
Jul 12 02:05:13.128: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 184, data seq: 2
Jul 12 02:05:15.265: PROCESS: INFO: MC[2] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 159, data seq: 1
Jul 12 02:05:15.265: PROCESS: INFO: BR[2] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 159, data seq: 1
Jul 12 02:05:15.265: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:15.265: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:15.267: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:15.267: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:15.267: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 184, data seq: 2
Jul 12 02:05:15.267: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 184, data seq: 2
Jul 12 02:05:21.764: PROCESS: WARNING: MC[2] Tcp connection to peer: 10.10.3.1, socket: 9 is reset,
 detected at TCP read handler
Jul 12 02:05:21.874: PROCESS: WARNING: MC[2] Tcp connection to peer: 10.10.4.1, socket: 7 is reset,
 detected at TCP read handler
Jul 12 02:05:23.756: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 306, data seq: 5
Jul 12 02:05:23.756: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 306, data seq: 5
Jul 12 02:05:26.806: PROCESS: INFO: MC[2] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 185, data seq: 3
Jul 12 02:05:26.806: PROCESS: INFO: BR[2] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 185, data seq: 3
Jul 12 02:05:26.807: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:26.807: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:05:26.807: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:26.807: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.0.0/16 from
10.20.1.1
Jul 12 02:05:26.807: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:26.807: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:26.807: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:05:26.808: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:26.808: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.0.0/16 from
10.20.1.1
Jul 12 02:05:26.808: PROCESS: INFO: MC[2] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:26.808: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:26.810: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:05:26.810: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.0.0/16 from
10.20.1.1
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
```

```
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.0.0/16 from
10.20.1.1
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:26.811: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 299, data seq: 4
Jul 12 02:05:26.811: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 299, data seq: 4
Jul 12 02:05:29.366: PROCESS: INFO: MC[3] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 194, data seq: 3
Jul 12 02:05:29.366: PROCESS: INFO: BR[3] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 194, data seq: 3
Jul 12 02:05:29.366: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.2/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.2/32 from
10.20.1.1
Jul 12 02:05:29.367: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:29.369: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:05:29.370: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:29.371: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:29.372: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:05:29.372: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.2/32 from
10.20.1.1
Jul 12 02:05:29.372: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:05:29.372: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:05:29.372: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:05:29.372: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.2.1/32 from
10.20.1.1
Jul 12 02:05:29.372: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:05:29.373: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.2/32 from
10.20.1.1
Jul 12 02:05:29.373: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 300, data seq: 4
```

```
Jul 12 02:05:29.374: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 300, data seq: 4
Jul 12 02:05:32.997: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 297, data seq: 5
Jul 12 02:05:32.997: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.30.1.1,
 data size: 297, data seq: 5
Jul 12 02:05:48.848: PROCESS: INFO: MC[2] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 299, data seq: 5
Jul 12 02:05:48.848: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 299, data seq: 5
Jul 12 02:05:48.847: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 300, data seq: 5
Jul 12 02:05:48.847: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 300, data seq: 5
Jul 12 02:05:52.133: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.10.1.1,
 data size: 258, data seq: 9
Jul 12 02:05:52.134: PROCESS: INFO: MC[2] SAF peering publish sub-service: Capability, origin:
10.10.1.1, size: 548, compressed size: 238, publish seq: 9, publish reason: On-Demand, result: Peering
 Success

Jul 12 02:07:46.213: PROCESS: INFO: BR[2] SAF peering received sub-service: Capability, from: 10.10.1.1,
 data size: 288, data seq: 10
Jul 12 02:07:46.213: PROCESS: INFO: MC[2] SAF peering publish sub-service: Capability, origin:
10.10.1.1, size: 822, compressed size: 268, publish seq: 10, publish reason: On-Demand, result:
Peering Success

Jul 12 02:08:27.105: PROCESS: INFO: MC[3] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 183, data seq: 6
Jul 12 02:08:27.105: PROCESS: INFO: BR[3] SAF peering received sub-service: site-prefix, from:
10.20.1.1, data size: 183, data seq: 6
Jul 12 02:08:27.107: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:08:27.107: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:08:27.107: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:08:27.107: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:08:27.107: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:08:27.107: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:08:27.107: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:08:27.106: PROCESS: INFO: MC[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:08:27.106: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:08:27.106: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:08:27.106: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:08:27.106: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:08:27.109: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.1.1/32 from
10.20.1.1
Jul 12 02:08:27.109: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.1/32 from
10.20.1.1
Jul 12 02:08:27.109: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.0.0/24 from
10.20.1.1
Jul 12 02:08:27.109: PROCESS: INFO: BR[3] SAF peering site prefix update: prefix 10.20.3.1/32 from
10.20.1.1
Jul 12 02:08:27.110: PROCESS: INFO: MC[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 265, data seq: 7
```

```
Jul 12 02:08:27.110: PROCESS: INFO: BR[3] SAF peering received sub-service: Capability, from: 10.20.1.1,
 data size: 265, data seq: 7
Jul 12 02:08:47.153: PROCESS: INFO: BR[2] Deregister CMD IDB : idb Tunnel1, client id 4, result
Succeed
Jul 12 02:08:47.155: PROCESS: INFO: BR[2] Deregister CMD IDB : idb Tunnel10, client id 4, result
Succeed
Jul 12 02:08:47.157: PROCESS: INFO: BR[2] Deregister CMD IDB : idb Tunnel30, client id 4, result
Succeed
Jul 12 02:08:47.174: PROCESS: INFO: BR[2] SAF peering destroyed: 10.10.1.1 to 10.10.1.1
Jul 12 02:08:47.177: PROCESS: INFO: MC[2] SAF peering listener destroyed: 10.10.1.1
Jul 12 02:08:47.177: PROCESS: INFO: BR[2] SAF peering unsubscribe sub-service: pmi succeed
Jul 12 02:08:47.178: PROCESS: INFO: MC[2] SAF peering unsubscribe sub-service: cent-policy succeed
Jul 12 02:08:47.178: PROCESS: INFO: BR[2] SAF peering unsubscribe sub-service: site-prefix succeed
Jul 12 02:08:47.178: PROCESS: INFO: MC[2] SAF peering unsubscribe sub-service: site-prefix succeed
Jul 12 02:08:47.178: PROCESS: INFO: BR[2] SAF peering unsubscribe sub-service: globals succeed
Jul 12 02:08:47.178: PROCESS: INFO: MC[2] SAF peering unsubscribe sub-service: Capability succeed
Jul 12 02:08:47.179: PROCESS: INFO: BR[2] SAF peering unsubscribe sub-service: Capability succeed
Jul 12 02:08:47.179: PROCESS: INFO: MC[2] SAF peering unsubscribe sub-service: globals succeed
Jul 12 02:08:47.179: PROCESS: INFO: BR[2] Eigrp uncfg opcode: Neighbor multihop, vrf: , interface:
Loopback1, action: Delete, result: Succeed
Jul 12 02:08:47.226: PROCESS: INFO: BR[2] stopped
Jul 12 02:08:47.226: PROCESS: INFO: MC[2] SAF peering unsubscribe sub-service: pmi succeed
Jul 12 02:08:48.234: PROCESS: INFO: MC[2] Eigrp uncfg opcode: Family only, vrf: , interface: Loopback1,
 action: Delete, result: Succeed
Jul 12 02:08:48.246: PROCESS: INFO: BR[3] Deregister CMD IDB : idb Tunnel0, client id 4, result
Succeed
Jul 12 02:08:48.248: PROCESS: INFO: BR[3] Deregister CMD IDB : idb Tunnel11, client id 4, result
Succeed
Jul 12 02:08:48.249: PROCESS: INFO: BR[3] Deregister CMD IDB : idb Tunnel31, client id 4, result
Succeed
Jul 12 02:08:48.261: PROCESS: INFO: BR[3] SAF peering destroyed: 10.10.1.1 to 10.10.1.1
Jul 12 02:08:48.263: PROCESS: INFO: MC[3] SAF peering listener destroyed: 10.10.1.1
Jul 12 02:08:48.264: PROCESS: INFO: BR[3] SAF peering unsubscribe sub-service: pmi succeed
Jul 12 02:08:48.264: PROCESS: INFO: MC[3] SAF peering unsubscribe sub-service: cent-policy succeed
Jul 12 02:08:48.263: PROCESS: INFO: BR[3] SAF peering unsubscribe sub-service: site-prefix succeed
Jul 12 02:08:48.263: PROCESS: INFO: MC[3] SAF peering unsubscribe sub-service: site-prefix succeed
Jul 12 02:08:48.263: PROCESS: INFO: BR[3] SAF peering unsubscribe sub-service: globals succeed
Jul 12 02:08:48.263: PROCESS: INFO: MC[3] SAF peering unsubscribe sub-service: Capability succeed
Jul 12 02:08:48.263: PROCESS: INFO: BR[3] SAF peering unsubscribe sub-service: Capability succeed
Jul 12 02:08:48.263: PROCESS: INFO: MC[3] SAF peering unsubscribe sub-service: globals succeed
Jul 12 02:08:48.263: PROCESS: INFO: BR[3] Eigrp uncfg opcode: Neighbor multihop, vrf: , interface:
Loopback2, action: Delete, result: Succeed
Jul 12 02:08:48.298: PROCESS: INFO: BR[3] Deregister CMD client : client id 4, result Succeed
Jul 12 02:08:48.299: PROCESS: INFO: BR[3] stopped
Jul 12 02:08:48.299: PROCESS: INFO: MC[3] SAF peering unsubscribe sub-service: pmi succeed
Jul 12 02:08:49.301: PROCESS: INFO: MC[3] Eigrp uncfg opcode: Family only, vrf: , interface: Loopback2,
 action: Delete, result: Succeed
```

Displays event trace for PfRv3 processes.

Additional References for PfRv3 Event Tracing

Related Documents

Related Topic	Document Title
PfRv3 commands	Cisco IOS Performance Routing Version 3 Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for PfRv3 Event Tracing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 346: Feature Information for PfRv3 Event Tracing

Feature Name	Releases	Feature Information
Event Trace for PFRv3 Errors and PFRv3 Channels	Cisco IOS XE Fuji 16.9.1	The following commands were introduced or modified: **show monitor event-trace pfrv3 sub-comp channel**, **show monitor event-trace pfrv3 sub-comp pdp**, **show monitor event-trace pfrv3 sub-comp policy**, **show monitor event-trace pfrv3 sub-comp process**.

CHAPTER 290

PfRv3 Command References

The following tables lists various Cisco IOS commands that are used for PfRv3 along with the command mode from which they are entered.

Table 347: Configuration Commands for PfRv3

Command mode	Command name	Description
Interface configuration	**bandwidth** *bandwidth-value*	Configures inherited and received bandwidth values for the tunnel interface. The bandwidth value is in kilobits and the valid values are 1 to 10000000.
Border configuration	**border**	Defines a device as a border.
Domain master controller configuration	**branch-to-branch**	Disable branch to branch PfR optimization. This is configured on Branch Masters. **Note** Configuring the command results in two different bahaviors for the different releases. See, PfRv3 Command Reference guide.
Configuration	**domain** *domain name*	Configures a top level domain for PfRv3.
Configuration interface	**domain** *domain name* **path***path-name*	Configures a path for the domain for PfRv3.
Global configuration	**domain** *domain name* **path***path-name* **path-id***path-id*	Configures a path and path-id for a specified domain.
Global configuration	**domain path** *isp-name* **zero-sla**	Configures Zero SLA on tunnel interface for an ISP path.
Master controller configuration	**hub** *ip-address*	Configures an IP address for the hub.
Domain class configuration	**match {application | dscp}**	Specifies applications or DSCP policies to be associated with a class.

Command mode	Command name	Description
Master controller configuration	**master** *ip-address*	Configures an IP address for the master controller.
Domain VRF configuration	**master {hub | branch | transit}**	Defines a device as a master type. You can configure a master device as a hub, border, or a transit.
Domain VRF configuration	**master transit** *pop-id*	Configures an ID for the master transit branch.
Domain-class configuration	**path-preference** *path-name* **fallback** *path-name*	Specifies a path preference for a traffic class policy.
Master controller class type	**priority** *priority-number* **[jitter | loss | one-way-delay] threshold** *thershold-value*	Specifies threshold values for user-defined policies.
Domain configuration	**vrf** *vrf- name*	Configures a Virtual Routing and Forwarding (VRF) instance for a domain.

Table 348: Show Commands for PfRv3

Command mode	Command name	Description
Privileged EXEC	**show domain** *domain-name* **border peering**	Displays the border router peering status.
Privileged EXEC	**show domain** *domain-name* **border pmi | begin prefix-learn**	Displays the automatically learned site-prefix status information of the hub-border router.
Privileged EXEC	**show domain** *domain-name* **border status**	Displays the status of the border routers configured at the hub site.
Privileged EXEC	**show domain** *domain-name* **border site-prefix**	Displays the site-prefix status information of the hub- border router.
Privileged EXEC	**show domain** *domain-name* **border channels**	Displays channel information from the hub-border site.
Privileged EXEC	**show domain** *domain-name* **border parent route**	Displays the parent route information of a border channel.
Privileged EXEC	**show domain** *domain-name* **border channels parent route**	Displays the parent route information of a channel.
Privileged EXEC	**show domain** *domain-name* **master exits**	Displays the summary of the external interfaces configured at the hub site.
Privileged EXEC	**show domain** *domain-name* **master peering**	Displays the peering information of the hub-master controller.

Command mode	Command name	Description
Privileged EXEC	**show domain** *domain-name* **master discovered-sites**	Displays branch sites that are remotely connected to the hub site.
Privileged EXEC	**show domain** *domain-name* **master site-prefix**	Displays the site-prefix status information of the hub- master controller.
Privileged EXEC	**show platform pfrv3 rp active smart-probe**	Displays the PfRv3 smart probe status on a Cisco ASR 1000 Series Aggregation Services Router configured at the hub site.
Privileged EXEC	**show platform pfrv3 fp active smart-probe**	Displays the PfRv3 active smart probes status of a embedded-service-processor on Cisco ASR 1000 Series Aggregation Services Routers.
Privileged EXEC	**show platform hardware qfp active feature pfrv3 client global pfrv3-instance detail**	Displays the platform hardware information on a Cisco ASR 1000 Series Aggregation Services Routers.
Privileged EXEC	**show flow monitor type performance-monitor**	Displays the flow monitor information for passive performance monitoring on the egress interface of WAN.
Privileged EXEC	**show domain** *domain-name* **master traffic-classes summary**	Displays the summary information of all the traffic classes.
Privileged EXEC	**show domain** *domain-name* **master traffic-classes**	Displays the status information of the traffic class for the hub-master controller.
Privileged EXEC	**show domain** *domain-name* **master traffic-classes policy** *policy-name*	Displays the occurrence of performance issues in a policy traffic class.
Privileged EXEC	**show domain** *domain-name* **master channels**	Displays channel information from the hub site.
Privileged EXEC	**show domain** *domain-name* **master channels link-name** *path-name*	Displays channel status information and the unreachable threshold crossing alerts (TCA) and on demand export (ODE) instances on a hub-master controller.
Privileged EXEC	**show domain** *domain-name* **master channels dst-site-id** *destination-site-id*	Displays the details of destination site-ids configured with hub-master controller.
Privileged EXEC	**show domain** *domain-name* **default master site-capability**	Displays the capability information of master controller.

Table 349: Debug Commands for PfRv3

Command mode	Command name	Description
Privileged EXEC	**debug platform hardware qfp active feature pfrv3 client**	Enables PfRv3 Cisco Quantum Flow Processor (QFP) client debug logging.
Privileged EXEC	**debug platform hardware qfp active feature pfrv3 datapath**	Enables PfRv3 Cisco Quantum Flow Processor (QFP) data path debug logging.
Privileged EXEC	**debug platform hardware qfp active feature pfrv3 pal**	Enables debug logging for PfRv3 in the Cisco Quantum Flow Processor (QFP).
Privileged EXEC	**debug platform software pfrv3**	Enables PfRv3 platform debug commands.

PART XII

Radio Aware Routing

CHAPTER 291

Overview of Radio Aware Routing

Introduction

Tech adoption plays a vital role in driving optimization and efficiency across all sectors today, including the Defence Industry, state and local government for search and rescue, law enforcement, and disaster assessment. These disciplines require the right information, in the right place, at the right time and mobile ad hoc networks are emerging to address these needs. The RFC5578 defines a PPP-over-Ethernet (PPPoE) based mechanism for integrating IP routers and mobile radios in ad hoc networks, enabling faster convergence, more efficient route selection, and better performance for traffic that is sensitive to delays.

In large mobile networks, connections to the routing neighbors are often interrupted due to distance and radio obstructions. When these signals do not reach the routing protocols, protocol timers are used to update the status of a neighbor. However, routing protocols have lengthy timers, which is not recommended in mobile networks.

Radio-Aware Routing (RAR) is a mechanism that uses radios to interact with the routing protocol OSPFv3 and EIGRP to signal the appearance, disappearance, and link conditions of one-hop routing neighbors. It addresses several of the challenges faced when merging IP routing and radio communications in mobile networks, especially those exhibiting mobile ad hoc (MANET) behaviour.

Mobile Ad Hoc Networking (MANET)

Mobile ad hoc networks are emerging as a means for delivering the benefits of IP networking to users operating beyond the reach of a fixed network. In ad hoc networks, mobile nodes associate on an extemporaneous or ad hoc basis. Ad hoc networks have numerous distinguishing characteristics when compared to conventional networking solutions:

- **`Self-forming`** — Nodes that come within radio range of each other can establish a network association without any pre-configuration or manual intervention.
- **`Self-healing`** — Nodes can join or leave rapidly without affecting operation of the remaining nodes.
- **`No infrastructure`** — In an ad hoc network, mobile nodes form their own network, and essentially become their own infrastructure.
- **`Peer to peer`** — Traditional networks typically support end systems operating in client-server mode. In an ad hoc network, mobile nodes can communicate and exchange information without prior arrangement and without reliance on centralized resources.
- **`Predominantly wireless`** — Historically networks have been mostly wired, and enhanced or extended through wireless access. The ad hoc environment is essentially wireless, but can be extended to support wired resources.

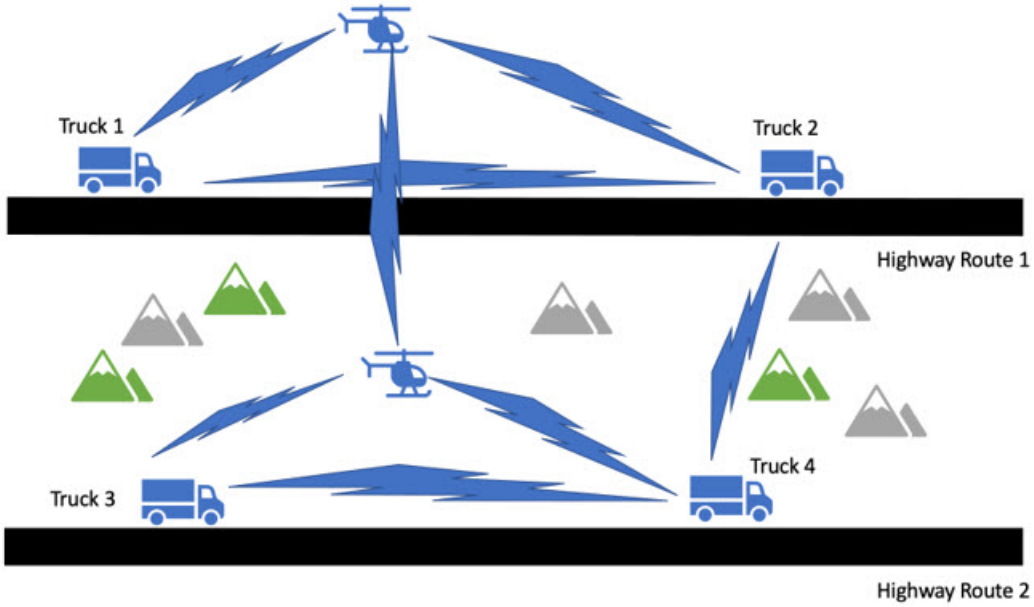
- **`Highly dynamic`** — Mobile nodes are in continuous motion and ad hoc networking topologies are constantly changing.

Collectively, these characteristics will enable ad hoc networks to deliver timely information to a new and underserved class of users. Ad hoc networking solutions can be applied to virtually any scenario that involves a cadre of highly mobile users or platforms (which may include stationary devices as well), a strong need to share IP-based information, and an environment in which fixed infrastructure is impractical, impaired, or impossible.

A Real-World Problem Description

The figure below shows a voice, video, data network between moving vehicles that consists of both ground and air vehicles, hence the network is mobile and it is a peer to peer mesh that changes as topographical obstructions are encountered. Networks with such topology are called mobile ad hock network, or MANET for short.

Figure 244: Mobile Ad-Hoc Network Topology



In the scenario in the drawing, all 4 trucks always have connectivity with the helicopters that are flying over the same road. The two helicopters always have line of sight and will always have a connection between each other. The trucks may even be able to connect to the other helicopter or a truck on the opposite road when conditions are favorable.

Here we see that the path between trucks 1 and 3 are completely blocked. The path between Truck 2 and 4 is about to be blocked.

Our existing routing protocols such as OSPFv3 and EIGRP need to adjust its path metrics very quickly to maintain a cohesive operational network. The routing protocol also needs a way to get that information from the radios and that requires a radio to router protocol that is delivered by Cisco Radio Aware Routing in the form of two open protocols:

- PPP over Ethernet (PPPoE)

- Dynamic Link Exchange Protocol (DLEP)

Both protocols are discussed later in this document.

- Feature Information for Radio Aware Routing, on page 3423
- Benefits of Radio Aware Routing, on page 3423

Feature Information for Radio Aware Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 350: Feature Information for Radio Aware Routing

Feature Name	Releases	Feature Information
Radio Aware Routing	Cisco IOS XE Release 17.8.1a	This feature was introduced for the following platforms: • Cisco Catalyst 8000V Edge Software
	Cisco IOS XE Release 17.11.1a	This feature was introduced for the the following platforms: • Cisco Catalyst 8200 Series Edge Platforms • Cisco Catalyst 8300 Series Edge Platforms • Cisco Catalyst 8500 Series Edge Platforms

Benefits of Radio Aware Routing

Radio Aware Routing offers the following benefits:

- Provides faster network convergence through immediate recognition of topographic obstructions and changes.
- Enables routing for failing or fading radio links.
- Allows easy routing between line-of-sight and non-line-of-sight paths.
- Provides faster convergence and optimal route selection so that traffic that is sensitive to delays, such as voice and video, is not disrupted.

- Provides efficient radio resources and bandwidth usage.
- Reduces impact on the radio links by performing congestion control in the router.
- Allows route selection based on radio power conservation.
- Enables decoupling of the routing and radio functionalities.
- Provides simple ethernet connection to PPPoE Extension and DLEP compliant radios.

CHAPTER 292

Overview of Dynamic Link Exchange Protocol

The Dynamic Link Exchange Protocol (DLEP) is a radio aware routing (RAR) protocol. DLEP provides a bidirectional, event-driven communication channel between the router and the radio to facilitate communication of changing link characteristics. In large mobile networks, connections to the routing neighbors are interrupted due to distance and radio obstructions. DLEP addresses the challenges faced when merging IP routing and radio frequency (RF) communications.

Benefits of DLEP

DLEP provides capabilities that enable:

- Optimal route selection based on feedback from radios
- Faster convergence when nodes join and leave the network
- Efficient integration of point-to-point, point-to-multipoint and broadcast multi-access radio topologies with multi-hop routing
- Flow-controlled communications between the radio and its partner router using rate-based Quality of Service (QoS) policies
- Dynamic shaping of fluctuating RF bandwidth in near real time to provide optimized use of actual RF bandwidth

This chapter contains the following sections:

Feature Information for Dynamic Link Exchange Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 351: Feature Information for Dynamic Link Exchange Protocol

Feature Name	Releases	Feature Information
Dynamic Link Exchange Protocol	Cisco IOS XE Release 17.8.1a	This feature was introduced for the following platforms: • Cisco Catalyst 8000V Edge Software
	Cisco IOS XE Release 17.11.1a	This feature was introduced for the the following platforms: • Cisco Catalyst 8200 Series Edge Platforms • Cisco Catalyst 8300 Series Edge Platforms • Cisco Catalyst 8500 Series Edge Platforms
IPv6 Unicast Support with DLEP	Cisco IOS XE Release 17.12.1a	The IPv6 Unicast Support feature introduces support for IPv6 data plane to RAR DLEP. This feature was introduced for the following platforms: • Cisco Catalyst 8000V Edge Software • Cisco Catalyst 8300 Series Edge Platforms • Cisco Catalyst 8500 and 8500L Series Edge Platforms

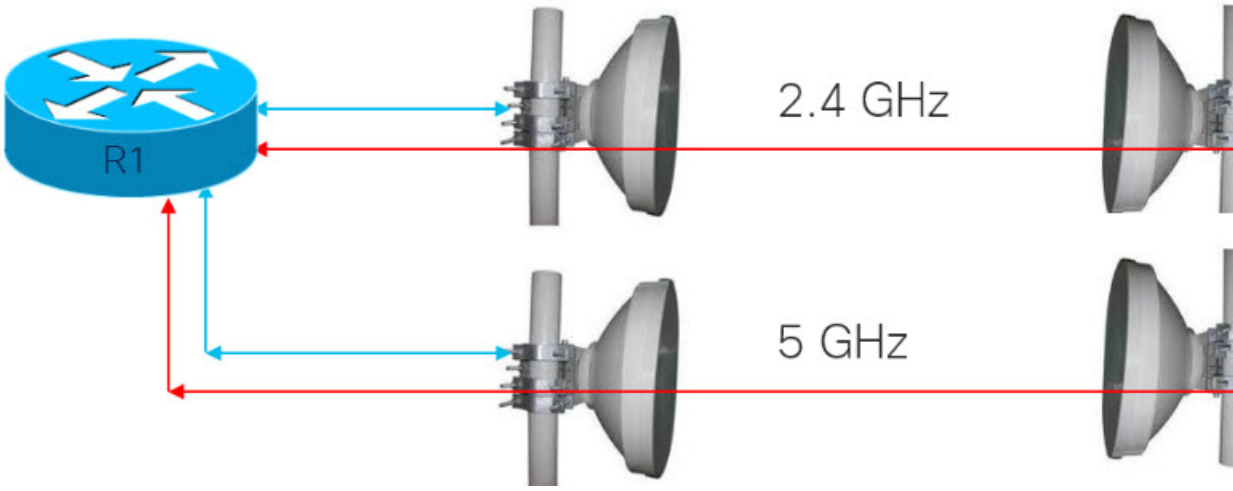
DLEP Topology

DLEP is a control protocol between a router and a DLEP-enabled radio modem. The DLEP message exchange between the router and radio allows the radio to communicate the router about the link quality. This is analogous to the way the bar icon on your cell phone indicates your Wi-Fi or LTE signal quality.

Figure 245: Network with DLEP

Using DLEP, we can make use of routing distances with equal cost, where metrics are updated in real time, based on the best path.

Figure 246: Network without DLEP



Without DLEP, there are two equal cost paths to any unadjusted routing protocol. With DLEP, routing metrics can be adjusted in real-time to favor the best path.

Note The final selection of a band is dependent on atmospheric conditions and interference.

Note in DLEP topology,

- the radio is the DLEP client,
- the router is the DLEP server, and
- the remote router is the DLEP neighbor.

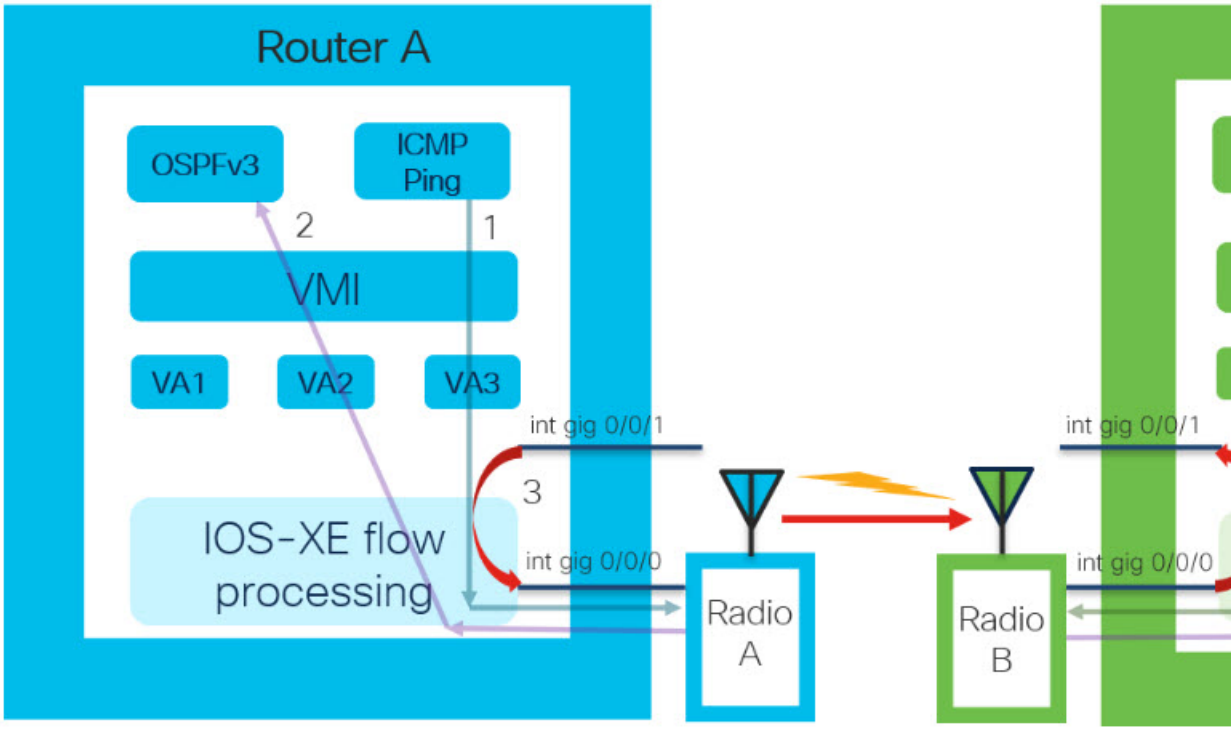
Interfaces in IOS-XE Platforms

- Virtual Multipoint Interface (VMI)
  - The VMI interface acts as an umbrella interface for all virtual access interfaces, per physical interface. VMI is used for routing protocols such as OSPFv3 and EIGRP, which will see a single VMI interface instead of all VA interfaces. This helps reduce routing table size without impacting the integrity of network.
- Virtual Template (VT)
  - Virtual Template serves as the template for every Virtual Access interface.
- Virtual Access (VA) interface
  - One VA is created for each DLEP Neighbor that is discovered on the network.
- Underlaying physical Layer 3 WAN interface (Gi 0/0/0 and Gi 0/0/1), or even sub-interface (Gi 0/0/0.2 or Gi0/0/1.2

Packet Flow Diagram with Flow Types

The following diagram describes the packet flow:

Figure 247: Packet Flow with Flow Types



Item	Description
1	Packet sent from IOS-XE to DLEP neighbor (to Radio): packet with DST MAC of neighbor MAC, neighbor IP sent out from DLEP Physical interface.
2	DLEP Packet (from radio) received thru DLEP Physical interface to IOS-XE: packet needs to be delivered to IOS routing protocol marked as from VMI interface.
3	End to End user data.

Prerequisites for DLEP

- DLEP requires the Network Advantage license.

Restrictions and Limitations

DLEP has the following restrictions and limitations:

- Multicast traffic is not supported with DLEP, but is supported with PPPOE.
- DLEP cannot be deployed with High Availability (HA) configuration.

- You must configure the VMI and Virtual-Template before attaching the Virtual-Template to a physical interface.
- Routers are connected over DLEP radio links, and only 1 radio per interface (VLAN or physical) is supported. The same interface cannot be used for connecting any other router (including remote DLEP peer router).
- You must remove all configurations for the virtual-template individually using the **no** form of the respective configuration commands, before removing the virtual-template using the **no interface virtual-template** command.
- You cannot change the configurations on the virtual-template and VMI interfaces while DLEP is enabled on the physical interface. To make such changes, disable DLEP by removing the DLEP configuration from the physical interface, make the changes, and re-configure DLEP on the physical interface.
- DLEP interface does not support jumbo frames (frames > 1500 bytes in size).
- Routing of internally generated application traffic (e.g. pingv6) with source as DLEP VMI / physical interface is not supported.
- You cannot use the **show ipv6 neighbor** command to view the information about DLEP neighbors.

Configuring DLEP

This section provides the following major sections for initiating, verifying, and managing all aspects of Dynamic Link Exchange Protocol (DLEP) on an interface. DLEP uses following interfaces that need to be configured:

- Physical interface
- VMI interface
- Virtual Templates

Important You must configure the VMI and Virtual-Template before attaching the Virtual-Template to a physical interface.

Note Routers are connected over DLEP radio links and only 1 radio per interface (VLAN or physical) is supported. The same interface cannot be used for connecting any other router (including remote DLEP peer router).

Configuring the Virtual Multipoint Interface

By default, virtual multipoint interfaces (VMIs) operate in aggregate mode, which means that all the virtual access interfaces created by DLEP sessions are aggregated logically under the configured VMI. Packets sent to the VMI are forwarded to the correct virtual access interface.

To configure the VMI, perform the following procedure:

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface vmi** *number*
	Example:
	`Router(config)# interface vmi 1`
	`Router(config-if)#`
		Creates a VMI and enters interface configuration mode.
		This example creates VMI1.

Step 4	**ip unnumbered** *interface* **Example:** `Router(config-if)#ip unnumbered gigabitEthernet 0/0/1`	Enables the IP on the VMI and bring it up without assigning a unique IP address to it.
Step 5	**physical-interface** *interface* **Example:** `Router(config-if)#physical-interface gigabitEthernet 0/0/0`	Binds the physical interface to VMI interface, for packet flow.
Step 6	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Enable ipv6 support under VMI interface.
Step 7	Configure routing protocols. **Example:** `Router(config-if)#ospfv3 1 ipv4 area 0`	Enable VMI interface to participate in OSPFv3 or EIGRP routing.

Step 8	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits the current mode.

Step 9	**router ospfv3 1** **Example:**	Global configuration for OSPFv3

	Command or Action	Purpose
	`Router(config)# router ospfv3 1`	

Step 10	**address-family ipv4 unicast**
	Example:
	`Router(config-router)# address-family ipv4 unicast`
	`Router(config-router-af)#`
		Adding address family for IPv4 unicast routing under global OSPFv3 configuration.

Configuring the Virtual Template

Configuring DLEP requires a virtual template to be defined. Perform this task to create the DLEP virtual template:

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface Virtual-Template** *number*
	Example:
	`Router(config)# interface Virtual-Template 1`
	`Router(config-if)#`
		Creates Virtual-Template interface and enters interface configuration mode.
		Note You need to use the same virtual-template interface for configuring DLEP on physical interface.

Step 4	**ip unnumbered** *interface* **Example:** `Router(config-if)#ip unnumbered gigabitEthernet 0/0/0`	Enables the IP on the VMI and bring it up without assigning a unique IP address to it.

Step 5	**ipv6 enable**
	Example:
	`Router(config-if)# ipv6 enable`
	`Router(config)#`
		Enables IPv6 support under Virtual-Template interface.

	Command or Action	Purpose

Step 6	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits the current mode.

Configuring the Physical Interface

DLEP configuration is currently supported on the WAN interface of routing platforms. As described above, you need to configure both the VMI and Virtual-Template interface before configuring the physical interface. There are various ways that DLEP configuration can be attached to WAN interface:

- DLEP template with well-known ip address [**Recommended**]
- DLEP template with TCP/UDP port based between server (Router) and client (Radio)
- DLEP template with dynamic port on server (Router)
- DLEP template attach in discovery mode

For each of the four modes mentioned above, the user also has the option to enable Generalized TTL Security Mechanism (GTSM).

To configure DLEP on an interface, perform the following procedure:

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface** *gi0/0/0 or gi0/0/1*
	Example:
	`Router(config)# interface gigabitEthernet 0/0/0`
	`Router(config-if)#`
		Enters interface configuration mode.

Step 4	**ipv6 enable** **Example:**	Enables IPv6 support under interface level.

	Command or Action	Purpose
	`Router(config-if)#ipv6 enable`	
Step 5	Assigning IP address to physical interface **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	Assigns physical IP address to the WAN interface.
Step 6	**ip dlep vtemplate** *port number* **Example:** `Router(config-if)#ip dlep vtemplate number 1`	Attaches DLEP Template to WAN interface, for discovery mode.
Step 7	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Brings up the interface.

Step 8	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits the current mode.

Configuring IPv6 with DLEP

From Cisco IOS XE 17.12.1a, DLEP can be configured with IPv6 dataplane.

Note When you configure DLEP for IPv6 traffic, enable IPv6 on the physical interface using the **ipv6 enable** command before associating the VMI interface with the physical interface using the **physical-interface <interface_name>** command.

Step 1 Enable IPv6 unicast routing:

```
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)#end
Router#
```

Step 2 Enable IPv6 on the physical interface:

```
Router#configure terminal
Router(config)# interface GigabitEthernet0/0/0
Router(config-if)# ipv6 enable
Router(config-if)#end
Router#
```

Step 3 Configure the Virtual Template Interface:

```
Router#configure terminal
Router(config)# interface virtual-template1
Router(config-if)# ip unnumbered GigabitEthernet0/0/0
Router(config-if)# ipv6 enable
Router(config-if)# ipv6 nd dad attempts 0
Router(config-if)#end
Router#
```

Step 4 Configure the Virtual Multipoint Interface:

```
Router#configure terminal
Router(config)# interface vmi1
Router(config-if)# ip unnumbered GigabitEthernet0/0/0
Router(config-if)# physical-interface GigabitEthernet0/0/0
Router(config-if)# ipv6 enable
Router(config-if)#end
Router#
```

Step 5 Configure the physical interface:

```
Router#configure terminal
Router(config)# interface GigabitEthernet0/0/0
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# ipv6 address 1000::1/64
Router(config-if)# ip dlep vtemplate 1
Router(config-if)#end
Router#
```

Attaching DLEP Virtual Templates

DLEP virtual templates can be attached in different modes to the WAN or sub-interface of the router.

Configuring DLEP Client/Server Based On Port Number

In this example, you are configuring the DLEP server, and client UDP and TCP ports.

Command or Action	Purpose

`Router(config)#interface gi0/0/0`
`Router(config-if)# ip address 10.0.0.1 255.255.255.0`
`Router(config-if)# ipv6 enable`
`Router(config-if)# ip dlep vtemplate 1 port 11113 tcp port 11114`
`client ip 10.0.0.3 port 11115`
`Router(config-if)# no shutdown`
	DLEP configuration where the server (router) is listening on UDP port 11113 and TCP port 11114, and the client (radio) is listening to TCP port 11115. The UDP port of the client is by default on 854.

Configuring DLEP with Dynamic Port on Server

In this example, you are configuring the DLEP server, and the UDP and TCP ports on the client.

Command or Action	Purpose
Router(config)#**interface gi0/0/0** Router(config-if)# **ip address 10.0.0.1 255.255.255.0** Router(config-if)# **ipv6 enable** Router(config-if)# **ip dlep vtemplate 1 client ip 10.0.0.3 port 11115** Router(config-if)# **no shutdown**	The DLEP configuration where the server (router) is listening to the default UDP or TCP ports, and the client (radio) is listening to TCP port 11115. The UDP port of the client is by default on 854.

Attaching DLEP Template in Discovery Mode

When in discovery mode, the DLEP server sends out Peer Discovery signals, and waits for a Peer Offer signal from the radio.

Command or Action	Purpose
Router(config)#**interface gi0/0/0** Router(config-if)# **ip address 10.0.0.1 255.255.255.0** Router(config-if)# **ipv6 enable** Router(config-if)# **ip dlep vtemplate 1** Router(config-if)# **no shutdown**	The DLEP configuration where the server (router) is listening to the default UDP or TCP ports.

Using a DLEP Template with a Well-Known IP Address

DLEP works based on RFC 8175, and uses the well-known IP address 224.0.0.117 on the server to communicate with radios.

Procedure

	Command or Action	Purpose
Step 1	**ip multicast-routing distributed** **Example:** **ip multicast-routing distributed**	Enables multicast routing on the router.
Step 2	**interface** *interface* **Example:** Router(config)# **interface gi0/0/0** Router(config-if)# **ip address 10.1.2.3 255.255.255.0** Router(config-if)# **no shutdown**	Configures IPv4 address of the server.
Step 3	**ip pim sparse-dense-mode** **Example:** Router(config-if)# **ip pim sparse-dense-mode**	Enables the PIM to operate in sparse or dense mode, depending on the multicast group.
Step 4	Configure forwarding **Example:**	Enables the router to forward multicast traffic.

	Command or Action	Purpose

	`Router(config-if)# ip mfib cef in`
	`Router(config-if)# ip mfib cef out`
	`Router(config-if)# ip mfib forwarding in`
	`Router(config-if)# ip mfib forwarding out`	

Step 5	**ip dlep vtemplate** <*Number*> **well-known ip** <*ip-address*> **Example:** `Router(config-if)# ip dlep vtemplate 1 well-known ip 224.0.0.117`	Enable sDLEP vTemplate to listen to multi-cast traffic. **Note** Under the command **show running-configuration** the output will display as "ip dlep vtemplate 1" which means "well-known ip 224.0.0.117" is hidden.

DLEP Quality of Service Configuration

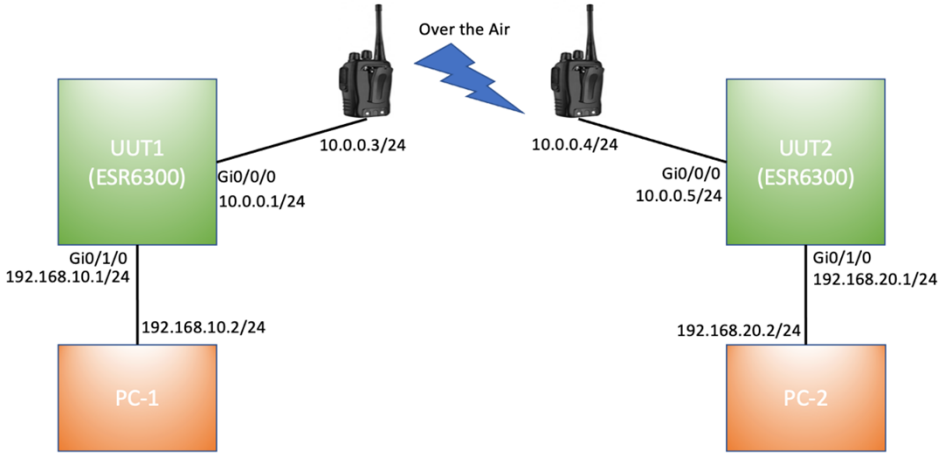
Quality of Service (QoS) for DLEP can be configured on the Virtual-Template which is associated with the physical interface. The data packets for DLEP flow through the Virtual Access interfaces that are created when DLEP neighbors come up.

Before proceeding, it is recommended to familiarize yourself with the Quality of Service (QoS) Configuration Guide for IOS-XE.

DLEP QoS Topology

The following figure shows a sample topology for DLEP with QoS.

Figure 248: QoS for DLEP



Based on above figure, the QoS policy is applied to egress of Virtual-Template attached to WAN interface Gi0/0/0.

Sample Configuration

```
Router# show running-config
Building configuration...

Current configuration : 7773 bytes
```

```
!
!
version 17.8
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
!
hostname UUT1
!
boot-start-marker
boot system bootflash:/c6300-universalk9.SSA.bin
!
ipv6 unicast-routing
!
class-map match-any CMAP_VIDEO
 match dscp 33
 match dscp 35
 match dscp 37
 match dscp 39
 match dscp af41
class-map match-any CMAP_VOICE
 match dscp 41
 match dscp 43
 match dscp 45
 match dscp 47
 match dscp 49
class-map match-any CMAP_DATA
 match dscp 9
 match dscp 11
 match dscp 13
 match dscp 15
 match dscp af11
!
policy-map Queue_Map
 class CMAP_VOICE
  bandwidth percent 40
  set dscp af11
 class CMAP_VIDEO
  bandwidth percent 50
 class CMAP_DATA
  bandwidth percent 10
  set dscp af23
!
interface Loopback1
 ip address 1.1.1.1 255.255.255.255
 ipv6 enable
 ospfv3 1 ipv4 area 0
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.0
 ip dlep vtemplate 1 port 11113 tcp port 11114 client ip 10.0.0.3 port 11115
 negotiation auto
 ipv6 address 1000::1/64
 ipv6 enable
!
interface GigabitEthernet0/0/1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/0
 switchport access vlan 30
```

```
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface Virtual-Template1
 ip unnumbered GigabitEthernet0/0/0
 ipv6 enable
 service-policy output Queue_Map
!
interface Vlan1
 no ip address
!
interface Vlan30
 ip address 192.168.10.1 255.255.255.0
 ipv6 address 1010::1/64
 ipv6 enable
 ospfv3 1 ipv6 area 0
!
interface Async0/2/0
 no ip address
 encapsulation scada
!
interface vmi1
 ip unnumbered GigabitEthernet0/0/0
 ipv6 address FE80::7E31:EFF:FE85:1E78 link-local
 ipv6 enable
 ospfv3 1 ipv4 area 0
 physical-interface GigabitEthernet0/0/0
!
router ospfv3 1
 !
 router-id 1.1.1.1
 address-family ipv4 unicast
 exit-address-family
!
end
Router#
```

Edit the Virtual-Template

Before you begin

To edit the Virtual-Template, you need to remove the configuration for **ip dlep vtemplate** on the WAN interface.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.

	Command or Action	Purpose

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface Virtual-Template** *number*
	Example:
	`Router(config)# interface Virtual-Template 1`
	`Router(config-if)#`
		Creates VMI interface and enters interface configuration mode.

Step 4	**Service-policy [input|output]** <*policy-map*> **Example:** `Router(config-if)# service-policy output Queue_Map`	Applies the policy-map to egress and ingress interfaces of the Virtual-Template.

Configuring DLEP on a Sub-Interface

DLEP can also be configured on a sub-interface. The following is an example:

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface** *interface*
	Example:
	`Router(config)# interface gi0/0/0`
	`Router(config-int)# no shut`
		Specifies an interface to configure.
		Note WAN interface must be in active state.

Step 4	**interface** *sub-interface* **Example:** `Router(config-if)# interface gi0/0/0.2`	Creates sub-interface gi0/0/0.2
Step 5	**Encapsulation dot1q** <*VLAN*> **native** **Example:** `Router(config-subif)# encapsulation dot1q 2 native`	Configures encapsulation dot1q over VLAN 2, and makes it native.

	Command or Action	Purpose
Step 6	**ip address** <*IP*> <*SUBNET*> **Example:** `Router(config-subif)# ip address 10.0.0.1 255.255.255.0`	Adds the IPv4 address for sub-interface.
Step 7	**ipv6 enable** **Example:** `Router(config-subif)# ipv6 enable`	Adds the IPv6 address for sub-interface. OSPFv3 IPv4 needs to have ipv6 support enabled on the interface level.

Step 8	**interface vmi** *number*
	Example:
	`Router(config-subif)# interface vmi 1`
	`Router(config-if)#`
		Creates VMI interface and enters interface configuration mode.

Step 9	**ip unnumbered** *interface* **Example:** `Router(config-if)# ip unnumbered gigibitEthernet 0/0/0.2`	Specifies VMI interface to use physical interface IP address.
Step 10	**physical-interface** *interface* **Example:** `Router(config-if)# physical-interface gigibitEthernet 0/0/0.2`	Binds the physical interface to VMI interface, for packet flow.
Step 11	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Enables IPv6 support under VMI interface.
Step 12	**interface** *sub-interface* **Example:** `Router(config-if)# interface gi0/0/0.2`	Configures the sub-interface for the DLEP template.

Step 13	**ip dlep vtemplate** <*number*> [*gtsm*] **client** {*ipv4* | *ipv6*}<*ip-address*> **port** *port*
	Example:
	`Router(config-subif)# ip dlep vtemplate 1 gtsm client 10.0.0`
	OR
	`Router(config-subif)# ip dlep vtemplate 1 port 11113 tcp port 11114 gtsm client ip 10.0.0.3 port`
		Attaches the DLEP Template to the sub-interface
		Note DLEP can be configured either in auto-discovery mode or manual configuration mode. Use the first example to configure DLEP in auto-discovery mode, or the second example to configure DLEP in manual mode.

Step 14	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits the current mode.

	Command or Action	Purpose

Step 15	**router eigrp** *<AS-NO>*
	Example:
	`Router(config)# router eigrp 1`
	`Router(config-router)#router-id 1.1.1.1`
	`Router(config-router)# network 10.0.0.0 0.0.0.255`
		Enables global configuration for EIGRP.

Configuring DLEP with OSPFv3

This section describes configuring DLEP with OSPFv3.

Note By default, OSPFv3 considers only the following DLEP metrics in route cost calculation:

- CDR
- RLQ
- MDR
- Resource

Step 1 Configure DLEP with IPv6:

See section Configuring IPv6 with DLEP.

Step 2 Configure the OSPF router:

```
Router#configure terminal
Router(config)#router ospfv3 1
Router(config-router)#router-id 200.200.200.200
Router(config-router)#address-family ipv4 unicast
Router(config-router-af)#end
Router#
```

Step 3 Configure OSPF on the VMI:

```
Router#configure terminal
Router(config)#interface vmi1
Router(config-if)#ospfv3 1 ipv4 area 0
Router(config-if)#ospfv3 1 ipv4 cost dynamic
Router(config-if)#ospfv3 1 ipv4 network manet
Router(config-if)#end
Router#
```

Configuring OSPFv3 for DLEP IPv6 unicast

From Cisco IOS XE 17.12.1, OSPFv3 for DLEP can be configured with IPv6 unicast.

Step 1 Configure DLEP with IPv6:

See section Configuring IPv6 with DLEP, on page 3434.

Step 2 Configure the OSPF router:

```
Router#configure terminal
Router(config)# router ospfv3 1
Router(config-router)# router-id 200.200.200.200
Router(config-router)# address-family ipv6 unicast
Router(config-router-af)#end
Router#
```

Step 3 Configure OSPF on the VMI:

```
Router#configure terminal
Router(config)# interface vmi1
Router(config-if)# ospfv3 1 ipv6 area 0
Router(config-if)# ospfv3 1 ipv6 cost dynamic              -> enables dynamic route cost.
Router(config-if)# ospfv3 1 ipv6 network manet             -> sets OSPF network type to MANET.
Router(config-if)#end
Router#
```

Configuring DLEP EIGRP

This section describes configuring DLEP with EIGRP.

Note By default, EIGRP considers only the following DLEP metrics in route cost calculation:

- CDR
- RLQ
- MDR

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 3	**interface vmi** *number*
	Example:
	`Router(config)# interface vmi 1`
	`Router(config-if)#`
		Creates VMI interface and enters interface configuration mode.

Step 4	**ip unnumbered** *interface* **Example:** `Router(config-if)# ip unnumbered gigibitEthernet 0/0/0`	Attaches the VMI interface with the physical interface IP address.
Step 5	**physical-interface** *interface* **Example:** `Router(config-if)# physical-interface gigibitEthernet 0/0/0`	Binds the physical interface to the VMI interface, for packet flow.
Step 6	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Enables IPv6 support under VMI interface.
Step 7	**no ip split-horizon eigrp** *number* **Example:** `Router(config-if)# no ip split-horizon eigrp 1`	Disables split-horizon mechanism.

Step 8	**exit**
	Example:
	`Router(config-if)# exit`
	`Router(config)#`
		Exits the current mode.

Step 9	**router eigrp** *autonomous-system-number*
	Example:
	`Router(config)# router eigrp 1`
	`Router(config-router)# router-id 1.1.1.1`
	`Router(config-router)# network 10.0.0.0 0.0.0.255`
	`Router(config-router)# passive-interface GigabitEthernet0/0/0`
		Enables EIGRP configuration with autonomous system number that identifies the services to the other EIGRP address-family routers.

Configuring EIGRP for DLEP IPv6 unicast

From Cisco IOS XE 17.12.1a, EIGRP for DLEP can be configured with IPv6 unicast.

Step 1 Configure the EIGRP router:

```
Router#configure terminal
Router(config)# ipv6 router eigrp 2
Router(config-rtr)# eigrp router-id 2.2.2.2
Router(config-rtr)#end
Router#
```

Step 2 Configure EIGRPv6 on VMI:

```
Router#configure terminal
Router(config)# interface vmi1
Router(config-if)# ipv6 eigrp 2
Router(config-if)# no ipv6 split-horizon eigrp 2
Router(config-if)#end
Router#
```

Optional Configurations for DLEP

There are a set of optional commands that are available to configure DLEP with a WAN or sub-interface. Using these commands, you can define the set of timeout intervals between peers, neighbors, and the heart-beat intervals for radios.

Command	Purpose
ip dlep set heartbeat-threshold ? <2-8> Threshold of missed heartbeat messages	Sets the heartbeat-threshold, between Server and Client.
ip dlep set peer-description ? LINE Peer Description Name	Defines the description with Peer.
ip dlep set peer-heartbeat-interval ? <1-60> Peer Heartbeat Interval timer duration in seconds	Sets the heartbeat interval between Server and client.
ip dlep set peer-discovery-interval ? <1-60> Peer Discovery Interval timer duration in seconds	Sets the peer discovery interval timer.

Removing the DLEP Configuration

Before you begin

You must remove all configurations for the virtual-template individually using the **no** form of the respective configuration commands, before removing the virtual-template using the **no interface virtual-template** command.

Procedure

	Command or Action	Purpose

Step 1	**enable**
	Example:
	`Router> enable`
	`Router#`
		Enables privileged EXEC mode.

	Command or Action	Purpose

Step 2	**configure terminal**
	Example:
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 3	**interface** *interface* **Example:** `Router(config)# interface gi0/0/0`	Specifies the interface.
Step 4	**no ip dlep vtemplate 1** **Example:** `Router(config-if)# no ip dlep vtemplate 1`	Clears the DLEP configuration. **Note** Disabling DLEP on the physical interface also removes all optional DLEP configurations indicated in section Optional Configurations for DLEP.
Step 5	**no int vmi** <*number*> **Example:** `Router(config-if)# no int vmi 1`	Removes the VMI interface on the router.

Step 6	**no int Virtual-Template** <*number*>
	Example:
	`Router(config-if)# no ip unnumbered gigabitEthernet 0/0/0`
	`Router(config-if)# no ipv6 enable`
	`Router(config-if)# no int Virtual-Template 1`
		Removing Virtual template on Router.

With the above configuration, DLEP will be removed from router. However, Virtual-Access interfaces that are created while bringing up DLEP neighbors, will still show up in the output of the **show ip interface brief** command until the system is rebooted.

Clearing DLEP Clients and Neighbors

This section describes how to clear DLEP configuration, using the **clear dlep client <interface> <peer id>** command, and possible ramifications.

The **clear dlep client** command clears the peer session.

The following shows an example:

Note First obtain the Peer ID from the output of the **show dlep client** command. Then, use that as the input to the **clear dlep client** command.

```
Router#show dlep client gi0/0/1

DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Local IP=15.0.0.10:55555 Sock=0
```

```
DLEP Local Radio IP=15.0.0.2:856 TCP Socket fd=1
 Peer ID=20, Virtual template=2
 Description: DLEP_Radio_2042
 Peer Timers (all values in milliseconds):
  Heartbeat=60000, Dead Interval=120000, Terminate ACK=240000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
Router#

Router#clear dlep client gi0/0/1 20
DLEP: Clear Client (peer) peer_id=20 from 15.0.0.10
```

There is another command that can be used to clear DLEP configuration. The **clear dlep neighbor <interface> <session id>** command clears DLEP neighbors. The session ID can be obtained using the **show dlep neighbor** command.

```
Router#clear dlep neighbor gi0/0/1 2215
DLEP: Clear neighbor sid=2215 from 195.0.0.2
```

DLEP Validation Commands

This section contains examples of how to verify the DLEP configuration on the router.

DLEP Configuration

Command	Information

`Router# show dlep config ?`
`  GigabitEthernet  GigabitEthernet IEEE 802.3z`
 	
`  |  Output modifiers`
`  <cr>  <cr>`
	DLEP configuration is supported only on WAN or sub-interface.

```
Router# show dlep config g0/0/1
DLEP Configuration for GigabitEthernet0/0/1

DLEP Peer Description -
DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:11117
DLEPv27 TCP Port = 11118
Virtual template=2
Timers (all values are in seconds):
Missed heartbeat threshold=2, Peer Terminate ACK timeout=10
Dlepv27 Applicable configs(in seconds):
Heartbeat interval=5, Discovery interval =5, Session Ack timeout=10
Neighbor activity timeout=0, Neighbor Down ACK timeout=10
```

DLEP Clients

Command	Information

Router# **show dlep clients ?**
  GigabitEthernet GigabitEthernet IEEE 802.3z
 	
  | Output modifiers
  <cr> <cr>
	DLEP clients is supported only on WAN or sub-interface.

```
Router# show dlep clients
DLEP Clients for all interfaces:


DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Server IP=10.0.0.1:11117 Sock=0 --> Local Router IP address


DLEP Client IP=10.0.0.2:859 TCP Socket fd=1 --> Directly connected Radio to the router
Peer ID=2, Virtual template=2
Description: DLEP_RadioSIM2
Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
Router#
```

DLEP Neighbor

Command	Information

Router# **show dlep neighbor** ?
  GigabitEthernet GigabitEthernet IEEE 802.3z
 	
  | Output modifiers
  <cr> <cr>
	DLEP neighbors is supported only on WAN or sub-interface.

```
Router# show dlep neighbor
DLEP Neighbors for all interfaces:

DLEP Neighbors for Interface GigabitEthernet0/0/1
DLEP Server IP=10.0.0.1:11117 Sock=0   ---> Local Router IP address

SID=2151  MAC_Address=a453.0e94.f861
  Addresses:
  IPv4 : 16.0.0.1          ---> Mac-Address and IPv4 address of neighbor's end-point device
Supported Metrics:
  RLQ RX Metric : 100
  RLQ TX Metric : 100
  Resources Metric : 100
```

```
  MTU Metric : 1500
  Latency Metric : 250 microseconds
  CDR RX Metric : 100000000 bps
  CDR TX Metric : 100000000 bps
  MDR RX Metric : 100000000 bps
  MDR TX Metric : 100000000 bps
```

DLEP Counters

Command	Information

`Router# show dlep counters ?`
`  GigabitEthernet  GigabitEthernet IEEE 802.3z`
 	
`  |  Output modifiers`
`  <cr>  <cr>`
	DLEP Counters is supported only on WAN or sub-interface, which will summarize port information, counters for peer, and neighbors.

```
UUT1# show dlep counters
DLEP Counters for GigabitEthernet0/0/1

Last Clear Time =

DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:11117
DLEPv5 TCP Port = 11118

Peer Counters:
RX Peer Discovery     0      TX Peer Offer             0
 RX Peer Offer         0      TX Peer Discovery         0
 RX Peer Init          0      TX Peer Init Ack          0
 RX Peer Init Ack      1      TX Peer Init              1
 RX Heartbeat          41     TX Heartbeat              41
 RX Peer Terminate     0      TX Peer Terminate Ack     0
 RX Peer Terminate Ack 0      TX Peer Terminate         0

Neighbor Counters:
RX Neighbor Up        1      TX Neighbor Up Ack        1
 RX Metric             0
 RX Neighbor Down      0      TX Neighbor Down Ack      0
 RX Neighbor Down Ack  0      TX Neighbor Down          1

Exception Counters:
RX Invalid Message    0      RX Unknown Message        0
 Pre-Existing Neighbor 0      Neighbor Resource Error   1
 Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
Peer Heartbeat Timer         41
 Peer Terminate Ack Timer     0
 Neighbor Terminate Ack Timer 0
 Neighbor Activity Timer      0
 Radio Connect Timer          5

Single Timer Wheel "Manet Infra Wheel"
  Granularity      = 250 msec
  Wheel size       = 4096
  Spoke index      = 3730
  Tick count       = 3423890
  Flags            = 0x00
  Active timers    = 1
  High water mark  = 1
  Started timers   = 171177
  Restarted timers = 2
```

```
 Cancelled timers = 5
 Expired timers   = 171169
 Long timers      = 0
 Long timer revs  = 0
 Timer suspends   = 0
```

Verifying DLEP Configuration

Use the following show commands to verify DLEP configuration.

Use the **show dlep clients** command to verify the DLEP client configuration based on port number:

```
Router# show dlep clients
DLEP Clients for all interfaces:

DLEP Clients for Interface GigabitEthernet0/0/0
DLEP Server IP=10.0.0.1:11113 Sock=0

DLEP Client IP=10.0.0.3:11115 TCP Socket fd=1
Peer ID=1, Virtual template=1
 Description: DLEP-Radio1-Path-1
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
```

Use the **show dlep clients** and **show dlep counters** commands to verify DLEP configuration with dynamic port on server:

```
Router# show dlep clients
DLEP Clients for all interfaces:

DLEP Clients for Interface GigabitEthernet0/0/0
DLEP Server IP=10.0.0.1:55555 Sock=0

DLEP Client IP=10.0.0.3:11115 TCP Socket fd=1
 Peer ID=1, Virtual template=1
 Description: DLEP-Radio1-Path-1
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
```

```
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
```

```
Router# show dlep counters
DLEP Counters for GigabitEthernet0/0/0

Last Clear Time =

DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:55555
DLEPv5 TCP Port = 55556

Peer Counters:
 RX Peer Discovery     0      TX Peer Offer             0
 RX Peer Offer         0      TX Peer Discovery         0
 RX Peer Init          0      TX Peer Init Ack          0
 RX Peer Init Ack      1      TX Peer Init              1
 RX Heartbeat          58     TX Heartbeat              58
 RX Peer Terminate     0      TX Peer Terminate Ack     0
 RX Peer Terminate Ack 0      TX Peer Terminate         0

Neighbor Counters:
 RX Neighbor Up        0      TX Neighbor Up Ack        0
 RX Metric             0
 RX Neighbor Down      0      TX Neighbor Down Ack      0
 RX Neighbor Down Ack  0      TX Neighbor Down          0

Exception Counters:
 RX Invalid Message    0      RX Unknown Message        0
 Pre-Existing Neighbor 0      Neighbor Resource Error   0
 Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
 Peer Heartbeat Timer         58
 Peer Terminate Ack Timer     0
 Neighbor Terminate Ack Timer 0
 Neighbor Activity Timer      0
 Radio Connect Timer          5

Single Timer Wheel "Manet Infra Wheel"
  Granularity      = 250 msec
  Wheel size       = 4096
  Spoke index      = 3592
  Tick count       = 3592
  Flags            = 0x00
  Active timers    = 1
  High water mark  = 2
  Started timers   = 164
  Restarted timers = 2
  Cancelled timers = 3
  Expired timers   = 158
  Long timers      = 0
  Long timer revs  = 0
  Timer suspends   = 0
```

Use the **show dlep clients** and **show dlep counters** commands to verify DLEP template attached in dscovery mode:

```
Router# show dlep clients
DLEP Clients for all interfaces:

DLEP Clients for Interface GigabitEthernet0/0/0
DLEP Server IP=10.0.0.1:55555 Sock=0

DLEP Client IP=10.0.0.3:11115 TCP Socket fd=1
```

```
 Peer ID=1, Virtual template=1
 Description: DLEP-Radio1-Path-1
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
```

```
Router# show dlep counters
DLEP Counters for GigabitEthernet0/0/0

Last Clear Time =

DLEP Version = RFC 8175
DLEP Server IP=10.0.0.1:55555
DLEPv5 TCP Port = 55556

Peer Counters:
 RX Peer Discovery     0      TX Peer Offer             0
 RX Peer Offer         3      TX Peer Discovery         194
 RX Peer Init          0      TX Peer Init Ack          0
 RX Peer Init Ack      3      TX Peer Init              3
 RX Heartbeat          710    TX Heartbeat              707
 RX Peer Terminate     0      TX Peer Terminate Ack     0
 RX Peer Terminate Ack 0      TX Peer Terminate         2

Neighbor Counters:
 RX Neighbor Up        0      TX Neighbor Up Ack        0
 RX Metric             0
 RX Neighbor Down      0      TX Neighbor Down Ack      0
 RX Neighbor Down Ack  0      TX Neighbor Down          0

Exception Counters:
 RX Invalid Message    0      RX Unknown Message        0
Pre-Existing Neighbor 0      Neighbor Resource Error   0
 Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
 Peer Heartbeat Timer         709
 Peer Terminate Ack Timer     2
 Neighbor Terminate Ack Timer 0
 Neighbor Activity Timer      0
 Radio Connect Timer          3

Single Timer Wheel "Manet Infra Wheel"
  Granularity      = 250 msec
  Wheel size       = 4096
  Spoke index      = 8
  Tick count       = 24584
  Flags            = 0x00
  Active timers    = 1
  High water mark  = 2
  Started timers   = 1209
  Restarted timers = 4
  Cancelled timers = 14
```

```
  Expired timers   = 1190
  Long timers      = 0
  Long timer revs  = 0
  Timer suspends   = 0
```

Use the **show dlep clients** command to verify DLEP template configuration with well-known IP address:

```
Router# show dlep clients
DLEP Clients for all interfaces:
DLEP Clients for Interface GigabitEthernet0/0/0
DLEP Server IP=10.1.2.3:55555 Sock=2

DLEP Client IP=10.1.2.4:854 TCP Socket fd=3
 Peer ID=1, Virtual template=1
 Description: OONF DLEP Radio
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link Latency Metric : 1000 microseconds
  Link CDR RX Metric : 104857600 bps
  Link CDR TX Metric : 104857600 bps
  Link MDR RX Metric : 104857600 bps
  Link MDR TX Metric : 104857600 bps
```

Use the **show policy-map interface Virtual-Access** command to verify QoS policy. In the following example, QoS policy is applied to Virtual-Template1, and data packets are flowing through Virtual-Access2 interface which is created when DLEP neighbors came up:

```
Router#show policy-map interface Virtual-Access 2
 Virtual-Access2
  Service-policy output: Queue_Map

    Class-map: CMAP_VOICE (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match:  dscp 41
      Match:  dscp 43
      Match:  dscp 45
      Match:  dscp 47
      Match:  dscp 49
      Queueing
      queue limit 208 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth 40% (400000 kbps)
      QoS Set
        dscp af11
          Marker statistics: Disabled

    Class-map: CMAP_VIDEO (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match:  dscp 33
      Match:  dscp 35
      Match:  dscp 37
      Match:  dscp 39
      Match:  dscp af41 (34)
      Queueing
```

```
    queue limit 208 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
    bandwidth 50% (500000 kbps)

  Class-map: CMAP_DATA (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match:  dscp 9
    Match:  dscp 11
    Match:  dscp 13
    Match:  dscp 15
    Match:  dscp af11 (10)
    Queueing
    queue limit 208 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 1024337/34827458
    bandwidth 10% (100000 kbps)
    QoS Set
      dscp af23
        Marker statistics: Disabled

  Class-map: class-default (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0000 bps, drop rate 0000 bps
    Match: any

    queue limit 208 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
```

Use the **show running-config** command to verify DLEP configuration on a sub-interface:

```
Router# show running-config
Building configuration...
Current configuration : 7726 bytes
!
!
version 17.8
hostname Router
!
boot-start-marker
boot system bootflash:/c6300-universalk9.SSA.bin
boot-end-marker
!
ipv6 unicast-routing
subscriber templating
!
license udi pid ESR-6300-CON-K9 sn FOC234304H3
license boot level network-advantage
!
interface Loopback1
 ip address 1.1.1.1 255.255.255.255
 ipv6 enable
 ospfv3 1 ipv4 area 0
!
interface GigabitEthernet0/0/0
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/0.10
 encapsulation dot1Q 10
 ip address 10.0.0.1 255.255.255.0
 ip dlep vtemplate 1 port 11113 tcp port 11114 client ip 10.0.0.2 port 11115
 ipv6 enable
```

```
!
interface Virtual-Template1
 ip unnumbered GigabitEthernet0/0/0.10
!
interface vmi1
 ip unnumbered GigabitEthernet0/0/0.10
 ipv6 address FE80::7E31:EFF:FE85:1E78 link-local
 ipv6 enable
 ospfv3 1 network manet
 ospfv3 1 ipv4 area 0
 physical-interface GigabitEthernet0/0/0.10
!
router ospfv3 1
 !
 address-family ipv4 unicast
 exit-address-family
 !
 address-family ipv6 unicast
 exit-address-family
!
end

Router#
```

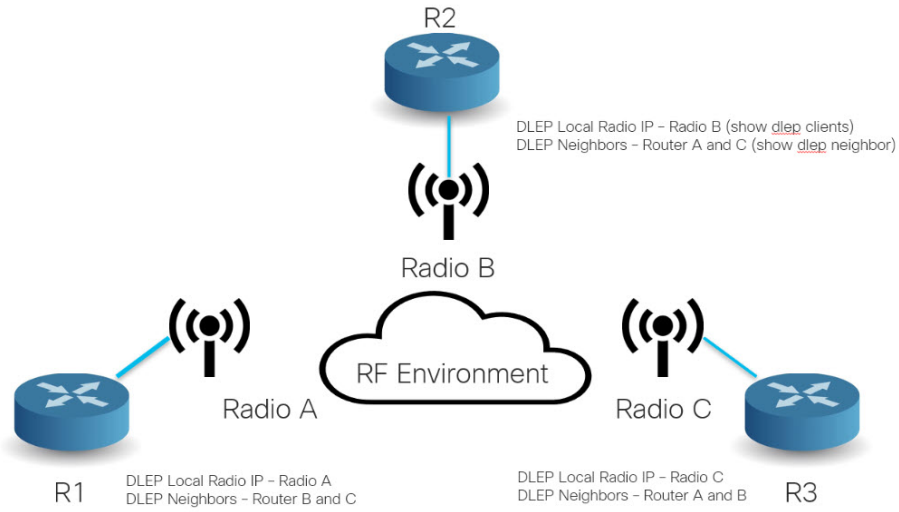
Use the **show dlep neighbors** command to verify IPv6 configuration with DLEP:

```
router# show dlep neighbors

DLEP Neighbors for all interfaces:
  DLEP Neighbors for Interface GigabitEthernet0/0/0.10
  DLEP Local IP=10.0.0.12:55555 Sock=0

SID=3323 Remote End-point MAC_Address=d478.9b5d.3800
Addresses:
  DLEP Remote IP : 10.0.0.2
  DLEP Remote IPv6 LL : FE80::D678:9BFF:FE5D:3800
```

Troubleshooting DLEP Configuration with show Commands

The following figure illustrates the sample topography that the various show commands in this section use.

Figure 249: Sample Installation



show DLEP Configuration

```
Router#     show dlep config Te0/0/0
DLEP Configuration for TenGigabitEthernet0/0/0

DLEP Peer Description -
DLEP Version = RFC 8175
DLEP Local IP=19.19.19.151:55113----> Local Router IP address
DLEPv27 TCP Port = 55114
 Virtual template=1
 Timers (all values are in seconds):
 Missed heartbeat threshold=2, Peer Terminate ACK timeout=10
 Dlepv27 Applicable configs(in seconds):
 Heartbeat interval=5, Discovery interval =5, Session Ack timeout=10
 Neighbor activity timeout=0, Neighbor Down ACK timeout=10
```

show DLEP Clients

```
Router# show dlep clients

DLEP Clients for all interfaces:

DLEP Clients for Interface TenGigabitEthernet0/0/0
DLEP Local IP=19.19.19.151:55113 Sock=2----> Local Router IP address


DLEP Local Radio IP=19.19.19.121:9121 TCP Socket fd=3----> Directly connected Radio IP
address
 Peer ID=42, Virtual template=1 ----> Attached virtual template for directly connected Radio
 from the router
Description: radio_9
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:----> Metrics from directly connected radio
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
```

```
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps

DLEP Clients for Interface TenGigabitEthernet0/0/1
DLEP Local IP=18.18.18.111:21115 Sock=0


DLEP Local Radio IP=18.18.18.112:8111 TCP Socket fd=1
 Peer ID=43, Virtual template=2
 Description: radio_7_8nw
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps

=====================================================
```

show DLEP Neighbors

```
Router#   show dlep neighbors

DLEP Neighbors for all interfaces:



DLEP Neighbors for Interface TenGigabitEthernet0/0/0
DLEP Local IP=19.19.19.151:55113 Sock=2----> Local Router IP address


SID=2187  Remote End-point MAC_Address=000c.2915.d4f8 ----> MAC address of end-point router
 interface
  Addresses:
  DLEP Remote IP : 19.19.19.161  DLEP Remote IPv6 LL : FE80::20C:29FF:FE15:D4F8
  Associated virtual access interface : Virtual-Access4----> DLEP Remote address and
link-local of end-point router
Supported Metrics:----> Supported Metrics to reach end point router directly connected
radio, based on the routing distance metrics will update apprropriately
  RLQ RX Metric : 100
  RLQ TX Metric : 100
  Resources Metric : 100
  MTU Metric : 1500
  Latency Metric : 250 microseconds
  CDR RX Metric : 100000000 bps
  CDR TX Metric : 100000000 bps
  MDR RX Metric : 100000000 bps
  MDR TX Metric : 100000000 bps

DLEP Neighbors for Interface TenGigabitEthernet0/0/1
DLEP Local IP=18.18.18.111:21115 Sock=0
```

```
SID=2188  Remote End-point MAC_Address=000c.2915.d402
  Addresses:
  DLEP Remote IP : 18.18.18.119  DLEP Remote IPv6 LL : FE80::20C:29FF:FE15:D402
  Associated virtual access interface : Virtual-Access3
 Supported Metrics:
  RLQ RX Metric : 100
  RLQ TX Metric : 100
  Resources Metric : 100
  MTU Metric : 1500
  Latency Metric : 250 microseconds
  CDR RX Metric : 100000000 bps
  CDR TX Metric : 100000000 bps
  MDR RX Metric : 100000000 bps
  MDR TX Metric : 100000000 bps
```

show DLEP Counters

```
Router# show dlep counters
DLEP Counters for TenGigabitEthernet0/0/0

Last Clear Time =

DLEP Version = RFC 8175
DLEP Local IP=19.19.19.151:55113----> Local Router IP address
DLEPv5 TCP Port = 55114

Peer Counters:
 RX Peer Discovery     0      TX Peer Offer             0
 RX Peer Offer         0      TX Peer Discovery         0
 RX Peer Init          0      TX Peer Init Ack          0
 RX Peer Init Ack      1      TX Peer Init              1
 RX Heartbeat          23     TX Heartbeat              23
 RX Peer Terminate     0      TX Peer Terminate Ack     0
 RX Peer Terminate Ack 0      TX Peer Terminate         0

Neighbor Counters:
 RX Neighbor Up        1      TX Neighbor Up Ack        1
 RX Metric             0
 RX Neighbor Down      0      TX Neighbor Down Ack      0
 RX Neighbor Down Ack  0      TX Neighbor Down          0

Exception Counters:
 RX Invalid Message    0      RX Unknown Message        0
 Pre-Existing Neighbor 0      Neighbor Resource Error   0
 Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
 Peer Heartbeat Timer         23
 Peer Terminate Ack Timer     0
 Neighbor Terminate Ack Timer 0
 Neighbor Activity Timer      0
 Radio Connect Timer          0


DLEP Counters for TenGigabitEthernet0/0/1

Last Clear Time =

DLEP Version = RFC 8175
DLEP Local IP=18.18.18.111:21115
DLEPv5 TCP Port = 21116

Peer Counters:
 RX Peer Discovery     0      TX Peer Offer             0
```

```
 RX Peer Offer         0      TX Peer Discovery         0
 RX Peer Init          0      TX Peer Init Ack          0
 RX Peer Init Ack      1      TX Peer Init              1
 RX Heartbeat          16     TX Heartbeat              16
 RX Peer Terminate     0      TX Peer Terminate Ack     0
 RX Peer Terminate Ack 0      TX Peer Terminate         0

Neighbor Counters:
 RX Neighbor Up        1      TX Neighbor Up Ack        1
 RX Metric             0
 RX Neighbor Down      0      TX Neighbor Down Ack      0
 RX Neighbor Down Ack  0      TX Neighbor Down          0

Exception Counters:
 RX Invalid Message    0      RX Unknown Message        0
 Pre-Existing Neighbor 0      Neighbor Resource Error   0
 Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
 Peer Heartbeat Timer         16
 Peer Terminate Ack Timer     0
 Neighbor Terminate Ack Timer 0
 Neighbor Activity Timer      0
 Radio Connect Timer          0

Single Timer Wheel "Manet Infra Wheel"
  Granularity      = 250 msec
  Wheel size       = 4096
  Spoke index      = 3078
  Tick count       = 470022
  Flags            = 0x00
  Active timers    = 2
  High water mark  = 3
  Started timers   = 4900
  Restarted timers = 10
  Cancelled timers = 105
  Expired timers   = 4783
  Long timers      = 0
  Long timer revs  = 0
  Timer suspends   = 0
```

Troubleshooting DLEP Configuration with debug Commands

This section shows two different troubleshooting scenarios.

Note It is recommended to use debug commands only under the guidance of Cisco TAC.

Scenario 1 : DLEP client is not reachable

In this senario, the router is not running in discovery mode, and the client/radio attributes have been explicitly configured.

Step 1: The output of **show dlep clients** indicates that there is no active client:

```
Router# show dlep clients

DLEP Clients for all interfaces:
```

```
DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Server IP=14.0.0.3:11117 Sock=-1
```

Step 2: Check the DLEP configuration:

```
Router#show dlep config
DLEP Configuration for GigabitEthernet0/0/1

DLEP Peer Description -
DLEP Version = RFC 8175
DLEP Server IP=14.0.0.3:11117
DLEPv27 TCP Port = 11118
 Virtual template=2
 Timers (all values are in seconds):
 Missed heartbeat threshold=2, Peer Terminate ACK timeout=10
 Dlepv27 Applicable configs(in seconds):
 Heartbeat interval=60, Discovery interval =5, Session Ack timeout=10
 Neighbor activity timeout=0, Neighbor Down ACK timeout=10

Router#show run int g0/0/1
Building configuration...

Current configuration : 245 bytes
!
interface GigabitEthernet0/0/1
 ip address 14.0.0.3 255.255.255.0
 ip dlep set peer-heartbeat-interval 60
 ip dlep vtemplate 2 port 11117 tcp port 11118 client ip 14.0.0.6 port 859
 negotiation auto
 ipv6 address 1111::1/120
 ipv6 enable
end
```

Step 3: Verify that the configuration on the radio (client) matches the configuration on the router (server) and that the router can reach the radio.

```
Router#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.199.184.2            9   0013.5f22.0b4a  ARPA   GigabitEthernet0/0/0
Internet  10.199.184.3            8   0018.7414.4e80  ARPA   GigabitEthernet0/0/0
Internet  10.199.184.19           -   a453.0e94.f638  ARPA   GigabitEthernet0/0/0
Internet  14.0.0.2                4   000c.297a.6b3d  ARPA   GigabitEthernet0/0/1
Internet  14.0.0.3                -   a453.0e94.f639  ARPA   GigabitEthernet0/0/1
Internet  14.0.0.6                0   Incomplete      ARPA

Router#ping 14.0.0.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 14.0.0.6, timeout is 2 seconds:
...
Success rate is 0 percent (0/3)
```

Step 4: The client ’14.0.0.6’ cannot be pinged. A quick check of the radio configuration revealed that the client IP address was actually 14.0.0.2.

```
Router#show run int g0/0/1
Building configuration...

Current configuration : 245 bytes
!
interface GigabitEthernet0/0/1
 ip address 14.0.0.3 255.255.255.0
 ip dlep set peer-heartbeat-interval 60
 ip dlep vtemplate 2 port 11117 tcp port 11118 client ip 14.0.0.6 port 859
 negotiation auto
```

```
 ipv6 address 1111::1/120
 ipv6 enable
end
```

Step 5: Correct the client IP address.

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int g0/0/1
Router(config-if)#no ip dlep vtemplate 2 port 11117 tcp port 11118 client ip 14.0.0.6 port
 859
Router(config-if)# ip dlep set peer-heartbeat-interval 60
Router(config-if)#ip dlep vtemplate 2 port 11117 tcp port 11118 client ip 14.0.0.6 port 859
Router(config-if)#^Z
Router#
*Feb 18 19:43:48.951: %SYS-5-CONFIG_I: Configured from console by console
```

Step 6: Verify the fix.

```
Router#show dlep counters
DLEP Counters for GigabitEthernet0/0/1

Last Clear Time =

DLEP Version = RFC 8175
DLEP Server IP=14.0.0.3:11117
DLEPv5 TCP Port = 11118

Peer Counters:
 RX Peer Discovery     0      TX Peer Offer             0
 RX Peer Offer         0      TX Peer Discovery         0
 RX Peer Init          0      TX Peer Init Ack          0
 RX Peer Init Ack      1      TX Peer Init              1
 RX Heartbeat          0      TX Heartbeat              0
 RX Peer Terminate     0      TX Peer Terminate Ack     0
 RX Peer Terminate Ack 0      TX Peer Terminate         0

Neighbor Counters:
 RX Neighbor Up        0      TX Neighbor Up Ack        0
 RX Metric             0
 RX Neighbor Down      0      TX Neighbor Down Ack      0
 RX Neighbor Down Ack  0      TX Neighbor Down          0

Exception Counters:
 RX Invalid Message    0      RX Unknown Message        0
 Pre-Existing Neighbor 0      Neighbor Resource Error   0
 Neighbor Not Found    0      Neighbor Msg Peer Not Up  0

Timer Counters:
 Peer Heartbeat Timer         0
 Peer Terminate Ack Timer     0
 Neighbor Terminate Ack Timer 0
 Neighbor Activity Timer      0
 Radio Connect Timer          1

Single Timer Wheel "Manet Infra Wheel"
  Granularity      = 250 msec
  Wheel size       = 4096
  Spoke index      = 1710
  Tick count       = 9902
  Flags            = 0x00
  Active timers    = 1
  High water mark  = 1
  Started timers   = 95
  Restarted timers = 4
```

```
  Cancelled timers = 4
  Expired timers   = 86
  Long timers      = 0
  Long timer revs  = 0
  Timer suspends   = 0

Router#
Router#show dlep clients

DLEP Clients for all interfaces:

DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Server IP=14.0.0.3:11117 Sock=0


DLEP Client IP=14.0.0.2:859 TCP Socket fd=1
 Peer ID=3, Virtual template=2
 Description: DLEP_RadioSIM2
 Peer Timers (all values in milliseconds):
  Heartbeat=60000, Dead Interval=120000, Terminate ACK=240000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
```

Scenario 2: DLEP session keeps timing out

In this scenario, the router is running in discovery mode.

Step 1: The DLEP session keeps flapping as indicated by the output of **show dlep client** sometimes shows an active client and sometimes it does not. Also, the VMI and virtual-access interfaces keep going up and down.

```
Router#show dlep clients

DLEP Clients for all interfaces:



DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Server IP=14.0.0.3:55555 Sock=0


DLEP Client IP=14.0.0.2:859 TCP Socket fd=1
 Peer ID=13, Virtual template=2
 Description: DLEP_RadioSIM2
 Peer Timers (all values in milliseconds):
  Heartbeat=5000, Dead Interval=10000, Terminate ACK=20000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
```

```
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
Router#

*Feb 18 20:01:32.577: %SYS-5-CONFIG_P: Configured programmatically by process Manet Infra
Background from console as console
*Feb 18 20:01:32.580: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3,
changed state to up
*Feb 18 20:01:32.584: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to up
*Feb 18 20:01:32.625: %LINEPROTO-5-UPDOWN: Line protocol on Interface vmi2, changed state
to up
Router#
Router#
*Feb 18 20:01:44.864: %LINEPROTO-5-UPDOWN: Line protocol on Interface vmi2, changed state
to down
*Feb 18 20:01:44.873: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3,
changed state to down
*Feb 18 20:01:44.878: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to down
*Feb 18 20:01:44.889: %SYS-5-CONFIG_P: Configured programmatically by process VTEMPLATE
Background Mgr from console as console
```

Step 2: Turn on the following debug commands to troubleshoot:

```
debug dlep server
debug dlep timer detail
debug dlep client error
debug dlep client infra
debug dlep client packet detail
debug dlep client state
```

Step 3: The debug logs indicate that the router/server sent a peer discovery signal and received a peer offer in return.

```
*Feb 18 20:14:59.553: dlepv27_encoder_signal_packet_start DLEP_SIGNAL_PEER_DISCOVERY(1)
*Feb 18 20:14:59.553: dlepv27_encoder_signal_packet_end tlv block size=0 packet length=8
*Feb 18 20:15:04.609: dlepv27_encoder_signal_packet_start DLEP_SIGNAL_PEER_DISCOVERY(1)
*Feb 18 20:15:04.609: dlepv27_encoder_signal_packet_end tlv block size=0 packet length=8
*Feb 18 20:15:04.611: dlepv27_decoder_signal_packet DLEP_SIGNAL_PEER_OFFER(2) data length
30
*Feb 18 20:15:04.611: dlepv27_decoder_peer_type_tlv DLEP_TLV_PEER_TYPE   flag - 0
dlepv27_decoder_parse_tlv_block last tlv 4; current block_len 11; next tlv 2
IPv4 Addr 14.0.0.2dlepv27_decoder_ipv4_conn_point_tlv DLEP_TLV_IPv4_CONN_POINT
dlepv27_decoder_parse_tlv_block last tlv 2; current block_len 0;
*Feb 18 20:15:04.611:
*Feb 18 20:15:04.611: dlepv27_decoder_packet rc(RC_DLEP_OK-0) state 0 signal 1 packet_len
38
*Feb 18 20:15:09.648: %DLEP_MSG-4-CONNECT_ERROR: TCP connect to Radio 14.0.0.2 failed via
Gi0/0/1. Error code: Resource temporarily unavailable
```

Step 4: The router/server sends a session initialization message, and receives an acknowledgement in return. The acknowledgement also carries the attributes of the radio/client. An examination of those attributes reveals that the heartbeat interval on the radio is set to 60 seconds.

```
*Feb 18 20:15:09.648: dlepv27_encoder_msg_packet_start DLEP_MSG_SESSION_INITIALIZATION(1)
*Feb 18 20:15:09.648: dlepv27_encoder_msg_packet_end tlv block size=13 packet length=17
*Feb 18 20:15:09.649:   Adding Peer for address 14.0.0.2(859), peer_id=22
*Feb 18 20:15:09.649: MANET_Infra: insert s=FFFF771137A8, type=2 (client insert)
*Feb 18 20:15:09.650: MANET Infra: Insert=FFFF745209B0 successful (client insert)
*Feb 18 20:15:09.650: MANET_Infra: insert s=FFFF771137A8, type=1 (client insert)
*Feb 18 20:15:09.650: MANET Infra: Insert=FFFF64C3CEE8 successful (client insert)
*Feb 18 20:15:09.650:  -0 Allocated peer context at 0xFFFF771137A8
```

```
*Feb 18 20:15:09.650: dlepv27_decoder_msg_packet DLEP_MSG_SESSION_INITIALIZATION_ACK(2)
data length 132
dlepv27_decoder_status_tlv DLEP_TLV_STATUS  status_code=0 desc ()
dlepv27_decoder_parse_tlv_block last tlv 1; current block_len 127; next tlv 4

*Feb 18 20:15:09.650: dlepv27_decoder_peer_type_tlv DLEP_TLV_PEER_TYPE   flag - 0
dlepv27_decoder_parse_tlv_block last tlv 4; current block_len 108; next tlv 5
dlepv27_decoder_heartbeat_interval_tlv DLEP_TLV_HEARTBEAT_INTERVAL  heartbeat=60000
dlepv27_decoder_parse_tlv_block last tlv 5; current block_len 100; next tlv 12

*Feb 18 20:15:09.650: dlepv27_decoder_latency_data_rate_value DLEP_TLV_LINK_MDR_METRIC_RX
 value=100000000
dlepv27_decoder_parse_tlv_block last tlv 12; current block_len 88; next tlv 13

*Feb 18 20:15:09.650: dlepv27_decoder_latency_data_rate_value DLEP_TLV_LINK_MDR_METRIC_TX
 value=100000000
dlepv27_decoder_parse_tlv_block last tlv 13; current block_len 76; next tlv 14

*Feb 18 20:15:09.650: dlepv27_decoder_latency_data_rate_value DLEP_TLV_LINK_CDR_METRIC_RX
 value=100000000
dlepv27_decoder_parse_tlv_block last tlv 14; current block_len 64; next tlv 15

*Feb 18 20:15:09.650: dlepv27_decoder_latency_data_rate_value DLEP_TLV_LINK_CDR_METRIC_TX
 value=100000000
dlepv27_decoder_parse_tlv_block last tlv 15; current block_len 52; next tlv 16

*Feb 18 20:15:09.650: dlepv27_decoder_latency_data_rate_value DLEP_TLV_LINK_LATENCY_METRIC
  value=250
dlepv27_decoder_parse_tlv_block last tlv 16; current block_len 40; next tlv 18
dlepv27_decoder_rlq_resource_value DLEP_TLV_LINK_RLQ_METRIC_RX  value=100
dlepv27_decoder_parse_tlv_block last tlv 18; current block_len 35; next tlv 19
dlepv27_decoder_rlq_resource_value DLEP_TLV_LINK_RLQ_METRIC_TX  value=100
dlepv27_decoder_parse_tlv_block last tlv 19; current block_len 30; next tlv 17
dlepv27_decoder_rlq_resource_value DLEP_TLV_LINK_RESOURCES  value=100
dlepv27_decoder_parse_tlv_block last tlv 17; current block_len 25; next tlv 20
dlepv27_decoder_mtu_tlv DLEP_TLV_LINK_MTU  mtu=100
dlepv27_decoder_parse_tlv_block last tlv 20; current block_len 19; next tlv 8
IPv4 Addr 14.0.0.2dlepv27_decoder_ipv4_address_tlv DLEP_TLV_IPV4_ADDRESS  operation=1
dlepv27_decoder_parse_tlv_block last tlv 8; current block_len 10; next tlv 10
IPv4 Subnet Addr 255.255.255.0dlepv27_decoder_ipv4_address_subnet_tlv
DLEP_TLV_IPV4_ATTACHED_SUBNET  operation=1 mask=24
dlepv27_decoder_parse_tlv_block last tlv 10; current block_len 0;
*Feb 18 20:15:09.651:
*Feb 18 20:15:09.651: dlepv27_decoder_packet rc(RC_DLEP_OK-0) state 1 signal 0 packet_len
136
Router#
```

Step 5: The router appears to be sending heartbeats 5 seconds apart:

```
*Feb 18 20:15:14.569: dlepv27_decoder_msg_packet DLEP_MSG_PEER_HEARTBEAT(16) data length 0


*Feb 18 20:15:14.569:  -curr_state Dlep In-Session State normalized_event=Dlep Peer Heartbeat
 Event  p2peer=0xFFFF771137A8  peer_id=22  p2neighbor=0x0
*Feb 18 20:15:14.569: dlepv27_decoder_packet rc(RC_DLEP_OK-0) state 2 signal 0 packet_len
4
Router#
*Feb 18 20:15:19.569: dlepv27_decoder_msg_packet DLEP_MSG_PEER_HEARTBEAT(16) data length 0


*Feb 18 20:15:19.569:  -curr_state Dlep In-Session State normalized_event=Dlep Peer Heartbeat
 Event  p2peer=0xFFFF771137A8  peer_id=22  p2neighbor=0x0
*Feb 18 20:15:19.569: dlepv27_decoder_packet rc(RC_DLEP_OK-0) state 2 signal 0 packet_len
4
Router#
```

Step6: The router is terminating the session, and receiving an acknowledgement of the same:

```
*Feb 18 20:15:24.569: dlepv27_decoder_msg_packet DLEP_MSG_SESSION_TERM(5) data length 5
dlepv27_decoder_status_tlv DLEP_TLV_STATUS  status_code=0 desc ()
dlepv27_decoder_parse_tlv_block last tlv 1; current block_len 0;
*Feb 18 20:15:24.569:
*Feb 18 20:15:24.569:  -curr_state Dlep In-Session State normalized_event=Dlep Peer Term
Event  p2peer=0xFFFF771137A8  peer_id=22  p2neighbor=0x0
*Feb 18 20:15:24.569:  -curr_state Dlep Terminating State normalized_event=Dlep Peer Term
ACK Event  p2peer=0xFFFF771137A8  peer_id=22  p2neighbor=0x0
*Feb 18 20:15:24.569: dlepv27_encoder_msg_packet_start DLEP_MSG_SESSION_TERM_ACK(6)
*Feb 18 20:15:24.569: dlepv27_encoder_msg_packet_end tlv block size=12 packet length=16
*Feb 18 20:15:24.570:  -curr_state Dlep Session Reset State normalized_event=Dlep Peer
sessoin reset Event  p2peer=0xFFFF771137A8  peer_id=22  p2neighbor=0x0
*Feb 18 20:15:24.570:  -0 Restart all peers on IDB GigabitEthernet0/0/1
*Feb 18 20:15:24.570: dlepv27_decoder_packet rc(RC_DLEP_OK-0) state 2 signal 0 packet_len
9
```

Step 7: An examination of the DLEP config reveals that the heartbeat on the router is set to 5 seconds:

```
Router#show dlep config
DLEP Configuration for GigabitEthernet0/0/1

DLEP Peer Description -
DLEP Version = RFC 8175
DLEP Server IP=14.0.0.3:55555
DLEPv27 TCP Port = 55556
 Virtual template=2
 Timers (all values are in seconds):
 Missed heartbeat threshold=2, Peer Terminate ACK timeout=10
 Dlepv27 Applicable configs(in seconds):
 Heartbeat interval=5, Discovery interval =5, Session Ack timeout=10
 Neighbor activity timeout=0, Neighbor Down ACK timeout=10
```

Step 8: Change the heartbeat to 60 seconds:

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int g0/0/1
Router(config-if)#no ip dlep
Router(config-if)#no ip dlep vtemplate
Router(config-if)#no ip dlep vtemplate 2
Router(config-if)#ip dlep set peer-heartbeat-interval 60
Router(config-if)# ip dlep vtemplate 2
Router(config-if)#^Z
```

Step 9: Verify the change fixed the problem:

```
Router#show dlep clients

DLEP Clients for all interfaces:


DLEP Clients for Interface GigabitEthernet0/0/1
DLEP Server IP=14.0.0.3:55555 Sock=0


DLEP Client IP=14.0.0.2:859 TCP Socket fd=1
 Peer ID=51, Virtual template=2
 Description: DLEP_RadioSIM2
 Peer Timers (all values in milliseconds):
  Heartbeat=60000, Dead Interval=120000, Terminate ACK=240000
 Neighbor Timers (all values in seconds):
  Activity timeout=0, Neighbor Down ACK=10

 Supported Metrics:
```

```
  Link RLQ RX Metric : 100
  Link RLQ TX Metric : 100
  Link Resources Metric : 100
  Link MTU Metric : 100
  Link Latency Metric : 250 microseconds
  Link CDR RX Metric : 100000000 bps
  Link CDR TX Metric : 100000000 bps
  Link MDR RX Metric : 100000000 bps
  Link MDR TX Metric : 100000000 bps
Router#
*Feb 18 20:38:03.708: %SYS-5-CONFIG_P: Configured programmatically by process Manet Infra
Background from console as console
*Feb 18 20:38:03.712: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3,
changed state to up
*Feb 18 20:38:03.716: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to up
*Feb 18 20:38:03.722: %LINEPROTO-5-UPDOWN: Line protocol on Interface vmi2, changed state
to up
```

Additional Debug Commands

Note It is recommended to use debug commands only under the guidance of Cisco TAC.

The following commands are available.

DLEP

```
debug dlep server detail
debug dlep timer detail
debug dlep neighbor error
debug dlep neighbor infrastructure detail
debug dlep neighbor infrastructure error
debug dlep neighbor metrics
debug dlep neighbor state
debug dlep neighbor all
debug dlep client error
debug dlep client infrastructure
debug dlep client packet dump
debug dlep client packet detail
debug dlep client state
```

VMI

```
debug vmi bma
debug vmi packet
debug vmi error
debug vmi multicast
debug vmi neighbor
debug vmi registries
```

Virtual Template

```
debug vtemplate cloning
debug vtemplate error
debug vtemplate  event
debug vtemplate subinterface
```

PPPOE

```
debug pppoe errors
debug pppoe events
debug pppoe packets
debug pppoe data
```

SSS

```
debug sss error
debug sss event
```

Related Documentation

Additional information can be found in the following resources:

Radio Aware Routing is discussed in this Cisco white paper.

Internet Engineering Task Force (IETF) RFC 8175

CHAPTER 293

Radio Aware Routing PPPoE

Radio-Aware Routing (RAR) is a mechanism that uses radios to interact with the routing protocols to signal the appearance, disappearance, and link conditions of routing neighbors.

Feature Information for RAR PPPoE

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 352: Feature Information for RAR PPPoE

Feature Name	Releases	Feature Information
IPv6 Multicast over RAR PPPoE	Cisco IOS XE Release 17.11.1a	This feature was introduced for the following platforms: • Cisco Catalyst 8000V Edge • Cisco Catalyst 8200 Series Edge Platforms • Cisco Catalyst 8300 Series Edge Platforms • Cisco Catalyst 8500 Series Edge Platforms
RAR PPPoE	Cisco IOS XE Release 17.8.1a	This feature was introduced for the following platforms: • Cisco Catalyst 8000V Edge
	Cisco IOS XE Release 16.6	This feature was introduced for the following platforms: • Cisco 4000 Series ISRs

Radio Aware Routing PPPoE Overview

This section provides a high-level description on how RAR, MANETs, and PPPoE work together.

About MANETs

Mobile Ad Hoc Networks (MANETs) for device-to-radio communications address the challenges faced when merging IP routing and mobile radio communications in ad hoc networking applications.

Through the device-to-radio link, the radio can inform the device immediately when a node joins or leaves, and this enables the device to recognize topology changes more quickly than if it had to rely on timers. Without this link-status notification from the radio, the device would likely time out while waiting for traffic. The link-status notification from the radio enables the device to respond faster to network topology changes. Metric information regarding the quality of a link is passed between the device and radio, enabling the device to more intelligently decide on which link to use.

With the link-status signaling provided by the device-to-radio link, applications such as voice and video work better because outages caused by topology changes are reduced or eliminated. Sessions are more stable and remain active longer.

PPPoE Extensions

PPPoE extensions are used when the router communicates with the radio. PPPoE is a well-defined communication mechanism between the client and the server. In the RAR implementation, radio takes the role of the PPPoE client and router takes the role of the PPPoE server. This allows a loose coupling of radio and router, while providing a well-defined and predictable communication mechanism. As PPPoE is a session or a connection oriented protocol, it extends the point-to-point radio frequency (RF) link from an external radio to an IOS-XE router.

Cross-layer feedback for device-to-radio integration of Radio-Aware Routing (RAR) takes advantage of the functions defined in RFC 5578. The RFC 5578 is an Internet Engineering Task Force (IETF) standard that defines PPP over Ethernet (PPPoE) extensions for Ethernet-based communications between a device and a mobile radio, that operates in a variable-bandwidth environment and has limited buffering capabilities. These extensions provide a PPPoE session-based mechanism for real time sharing of radio network status & link-quality metrics, and support credit-based flow control between router and RAR-compliant radio.

An RAR-compliant radio initiates a Layer 2 PPPoE session with its adjacent device on behalf of every device and radio neighbor discovered in the network. These Layer 2 sessions are the means by which radio network status for each neighbor link is reported to the device. The radio establishes the correspondence between each PPPoE session and each link to a neighbor.

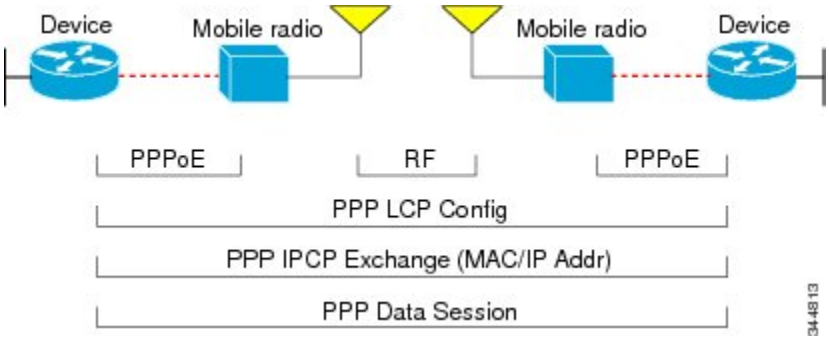
In the Cisco IOS implementation of PPPoE, each individual session is represented by virtual access interface (connectivity to a radio neighbor) on which, QoS can be applied with these PPPoE extensions.

PPPoE Interfaces for Mobile Radio Communications

The Mobile Ad Hoc Network (MANET) implementation uses PPP over Ethernet (PPPoE) sessions to enable intranodal communications between a device and its partner radio. Each radio initiates the PPPoE session as soon as the radio establishes a radio link to another radio. After the PPPoE sessions are active, a PPP session is established end-to-end (device-to-device). This is duplicated each time a radio establishes a new radio link. The virtual multipoint interface (VMI) on the device can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes. Underneath the VMI are virtual access interfaces that are associated with each of the PPP and PPPoE connections.

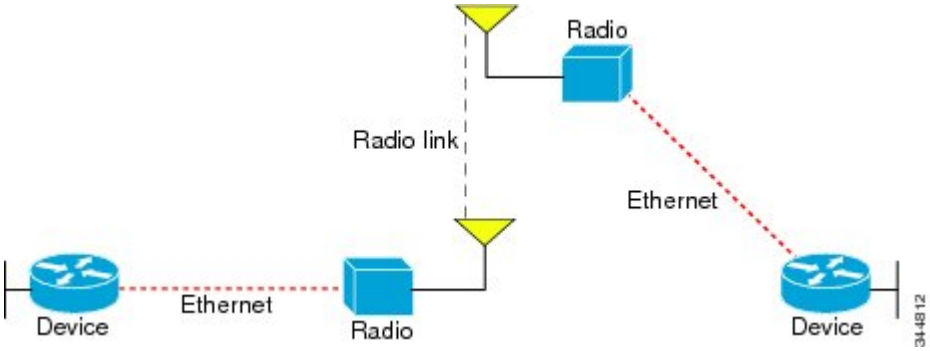
A PPPoE session is established between a device and a radio on behalf of every other device and radio neighbor located in the MANET. These Layer 2 sessions are the means by which radio network status gets reported to the Layer 3 processes in the device. The figure below shows the PPPoE session exchange between mobile devices and directional radios in a MANET network.

Figure 250: PPPoE Session Exchange Between Mobile Devices and Directional Radios



This capability requires that a Radio-Aware Routing (RAR)-compliant radio be connected to a device through Ethernet. The device always considers the Ethernet link to be up. If the radio side of the link goes down, the device waits until a routing update timeout occurs to declare the route down and then updates the routing table. The figure below shows a simple device-to-radio link topology.

Figure 251: Device-to-Radio Link



Neighbor Up and Down Signaling

Mobile Ad Hoc Networks (MANETs) are highly dynamic environments. Nodes might move into, or out of, radio range at a fast pace. Each time a node joins or leaves, the network topology must be logically reconstructed by the devices. Routing protocols normally use timer-driven hello messages or neighbor timeouts to track topology changes, but MANETs reliance on these mechanisms can result in unacceptably slow convergence.

The neighbor up/down signaling capability provides faster network convergence by using link-status signals generated by the radio. The radio notifies the device each time a link to another neighbor is established or terminated by the creation and termination of PPP over Ethernet (PPPoE) sessions. In the device, the routing protocols (Open Shortest Path First version 3 [OSPFv3] or Enhanced Interior Gateway Routing Protocol [EIGRP]) respond immediately to these signals by expediting formation of a new adjacency (for a new neighbor) or tearing down an existing adjacency (if a neighbor is lost). For example, if a vehicle drives behind a building and loses its connection, the device immediately senses the loss and establishes a new route to the vehicle through neighbors that are not blocked. This high-speed network convergence is essential for minimizing dropped voice calls and disruptions to video sessions.

When virtual multipoint interfaces (VMIs) with PPPoE are used and a partner node has left or a new one has joined, the radio informs the device immediately of the topology change. Upon receiving the signal, the device immediately declares the change and updates the routing tables. The signaling capability provides these advantages:

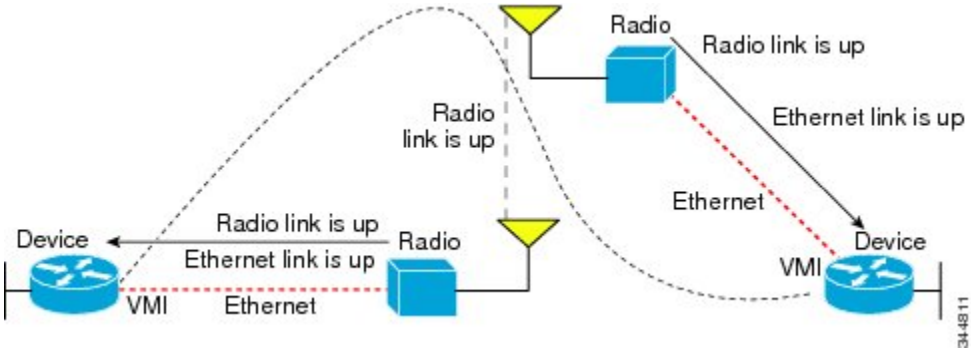
- Reduces routing delays and prevents applications from timing out.
- Enables network-based applications and information to be delivered reliably and quickly over directional radio links.
- Provides faster convergence and optimal route selection so that delay-sensitive traffic such as voice and video are not disrupted.
- Reduces impact on radio equipment by minimizing the need for internal queueing and buffering.
- Provides consistent quality of service for networks with multiple radios.

The messaging allows for flexible rerouting when necessary, in the following circumstances:

- Fading of the radio links
- Congestion of the radio links
- Radio link power fade
- Utilization of the radio

The figure below shows the signaling sequence that occurs when radio links go up and down:

Figure 252: Up and Down Signaling Sequence



PPPoE Credit-based and Metric-based Scaling and Flow Control

Each radio initiates a PPP over Ethernet (PPPoE) session with its local device as soon as the radio establishes a link to another radio. Once the PPPoE sessions are active for each node, a PPP session is then established end-to-end (device-to-device). This process is duplicated each time a radio establishes a new link.

The carrying capacity of each radio link might vary due to location changes or environmental conditions, and many radio transmission systems have limited buffering capabilities. To minimize the need for packet queueing in the radio, PPPoE protocol extensions enable the device to control traffic buffering in congestion situations. Implementing flow-control on these device-to-radio sessions allows use of quality of service (QoS) features such as fair queueing.

The flow-control solution utilizes a credit-granting mechanism documented in RFC 5578. When the PPPoE session is established, the radio can request a flow-controlled session. If the device acknowledges the request, all subsequent traffic must be flow controlled. If a flow-control session is requested and cannot be supported by the device, the session is terminated. Typically, both the radio and the device initially grant credits during session discovery. Once a device exhausts its credits, it must stop sending until additional credits are granted. Credits can be added incrementally over the course of a session.

Metrics scaling is used with high-performance radios that require high-speed links. The radio can express the maximum and current data rates with different scaler values. Credit scaling allows a radio to change the default credit grant (or scaling factor) of 64 bytes to its default value. You can display the maximum and current data rates and the scalar value set by the radio in the **show vmi neighbor detail** command output.

System Components

The Radio Aware Routing (RAR) feature is implemented using the MANET infrastructure comprising of different components such as PPPoE, Virtual Multipoint Interface (VMI), QoS, routing protocol interface and RAR protocols.

Virtual Multipoint Interface (VMI)

The VMI on the device can aggregate all of the per-neighbor PPPoE sessions from the radio Ethernet connection.

The VMI maps the sessions to appear to Layer 3 routing protocols and applications as a single point-to-multipoint,multiaccess, broadcast-capable network. However, the VMI preserves the integrity of the PPPoE sessions on the radio side so that each point-to-point connection can have its own quality of service (QoS) queue.

The VMI also relays the link-quality metric and neighbor up/down signaling from the radio to the routing protocols.The VMI signals are used by the Enhanced Interior Gateway Routing Protocol (EIGRP) for IPv4 and IPv6 neighbors and the Open Shortest Path First version 3 (OSPFv3) for IPv6 neighbors.

TheVMI can operate in two modes: bypass or aggregate.

Bypass Mode

This is the recommended mode for PPPoE in a MANET network.

In bypass mode, the virtual-access interfaces are directly exposed to applications running above L2. In bypass mode, you must still define a VMI because VMI continues to manage presentation of cross-layer signals, such as, neighbor up, neighbor down, and metrics. However, applications will still be aware of the actual underlying virtual-access interfaces and send packets to them directly.

If you are running multicast applications that require virtual-access interfaces to be exposed to applications above L2 directly, you can configure VMI to operate in bypass mode. Most multicast applications require that the virtual-access interfaces be exposed directly to routing protocols for the multicast Reverse Path Forwarding (RPF) to operate as expected.

Aggregate Mode

In this mode, all the virtual-access interfaces created by PPPoE sessions are aggregated logically under the configuredVMI. VMI on the router can aggregate multiple PPPoE sessions and multiplex them to look like a single interface to the routing processes. Applications above Layer 2 (L2), such as Enhanced Interior Gateway Routing Protocol (EIGRP) and OSPFv3, should be defined only on VMI. Underneath VMI are virtual access interfaces that are associated with each of the PPP/PPPoE connections. Packets sent to VMI are forwarded to the correct virtual-access interface(s).

Aggregate mode VMIs operate in Non-Broadcast Multiple Access (NBMA) mode. Multicast traffic is forwarded only to the NBMA neighbors where a listener for that group is present. This is the preferred mode when operating in PIM sparse mode.

Virtual Access Interface

The Virtual-Access interfaces are logically “underneath” the VMI interface. Each Virtual-Access interface representsa “destination” which is either a routing next-hop, or a multicast group. At the bottom of the interface hierarchy is the actual physical interface connecting the router and radio. The Virtual-Access interface funnels the traffic to the physical interface for transmission to the radio device.

PPPoE Packet Flow

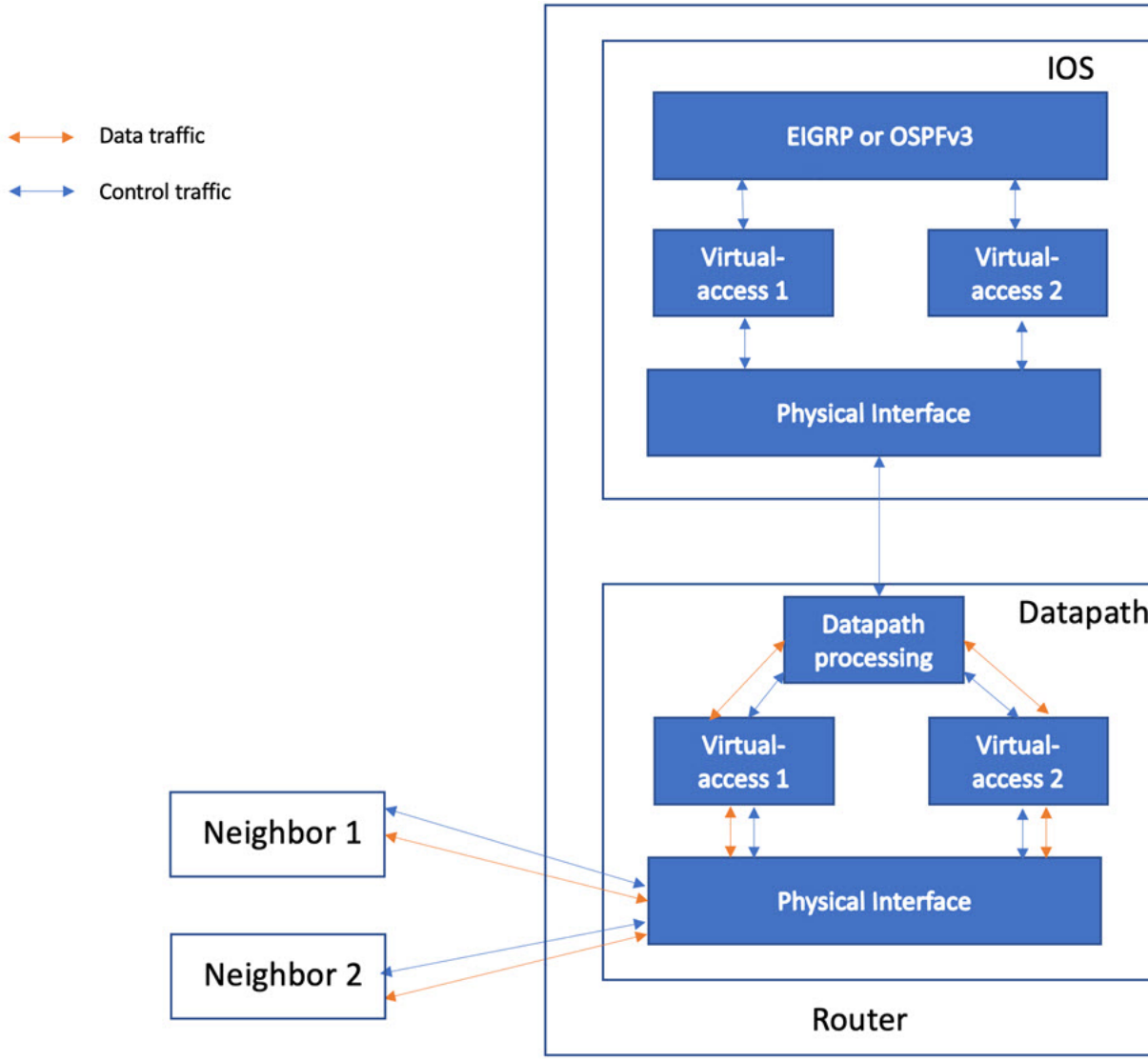
The Packet Flow diagram below illustrates the packet flow for both control and data packets over a PPPoE session when the VMI interface is in bypass mode.

All control traffic is sent to by the datapath to IOS where it is handed over to the appropriate protocol to be processed. The incoming interface for this traffic is the virtual-access interface associated with the neighbor

which is the source of the traffic. In this case, Virtual-access1 corresponds to Neighbor 1 and Virtual-Access 2 corresponds to Neighbor 2.

All data traffic is processed by the datapath and does not typically get sent to IOS.

Figure 253: Packet Flow



Restrictions

- Multicast traffic is not supported in aggregate mode.
- Cisco High Availability (HA) technology is not supported.

Enabling IPv6 Routing

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**ipv6 unicast-routing** **Example:** Router(config)# **ipv6 unicast-routing**	Enable IPv6 Unicast routing.
Step 3	ipv6 multicast-routing **Example:** Router(config)#**ipv6 multicast-routing**	Enables multicast routing on all IPv6-enabled interfaces, and enables multicast forwarding for PIM and MLD on all enabled interfaces of the device.

Creating a Subscriber Profile

The Subscriber Profile Support feature is functionality for the Subscriber Service Switch architecture, a Cisco IOS subsystem that connects subscribers to network access services at Layer 2. This functionality affects how the Subscriber Service Switch Manager determines a service for each subscriber with a combination of a policy and a service lookup model.

Note Configuring a subscriber profile for PPPoE service selection is required for VMI to function properly.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**subscriber templating** **Example:** Router(config)#**subscriber templating**	Configure subscriber templating.
Step 3	subscriber authorization enable **Example:** Router(config)#**subscriber authorization enable**	Enable Subscriber Service Switch type authorization.

Configuring PPPoE Service Policy

A service policy specifies class-based weighted fair queueing (CBWFQ). The class policies that make up the policy map are then applied to packets that satisfy the class map match criteria for the class.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**policy-map type service** *name* **Example:** Router(config)#**policy-map type service pppoe_rar**	Configure policy map type.
Step 3	**pppoe service** *name* **Example:** Router(config-service-policymap)# **pppoe service manet_radio**	Configure service policy map. **Note** Enter the PPPoE service policy name as manet_radio.

Configuring QoS Provisioning

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**policy-map** *name* **Example:** Router(config)#**policy-map rar_policer**	Create a policy map name.
Step 3	**class** *criteria_name* **Example:** Router(config-pmap)# **class class-default**	Configure policy criteria as system default class.
Step 4	**police** *target_rate* **conform-action** *action* **exceed-action** *action* **Example:**	Configure QoS policy for the Virtual Template Interface.

	Command or Action	Purpose
	Router(config-pmap-c)# **police 10000000 conform-action transmit exceed-action drop**	

Configuring PPPoE Service Selection

The PPPoE Service Selection feature uses service tags to enable a PPP over Ethernet (PPPoE) server to offer PPPoE clients a selection of services during call setup. The customer chooses one of the services offered, and the service is provided when the PPPoE session becomes active. This feature enables service providers to offer a variety of services and to charge customers according to the service chosen.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**bba-group pppoe** *name* **Example:** Router(config)# **bba-group pppoe rar_group_1**	Configure PPPoE global group.
Step 3	**virtual-template** *number* **Example:** Router(config-bba-group)# **virtual-template 1**	Attach the Virtual Template to the PPPoE bba group.
Step 4	**service profile** *name* **Example:** Router(config-bba-group)#**service profile pppoe_rar**	Attach the service policy name .

Configuring PPPoE on an Ethernet Interface

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *[type slot/port]* **Example:** Router(config)# **interface GigabitEthernet0/0/0**	Specifies an interface type and enters interface configuration mode.
Step 3	**ipv6 enable** **Example:** Router (config-if)# **ipv6 enable**	Enable IPv6 support under VMI interface. OSPFv3 IPv4 needs to have IPv6 support enabled on the interface level.
Step 4	**pppoe enable group** *group_name* **Example:** Router(config-if)#**pppoe enable group rar_group_1**	Enables PPPoE sessions on the interface or sub interface.

Configuring a Virtual Template Interface

You use the virtual template interface to dynamically clone configurations for each virtual access interface created for a virtual multipoint interface (VMI) neighbor. You can configure multiple virtual template interfaces for your VMI PPP over Ethernet (PPPoE) connections. The selection of which virtual template to use is predicated on the service name sent by the radio during PPPoE session establishment.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**interface Virtual-Template** *number* **Example:** Router(config)# **interface Virtual-Template 1**	Creates a Virtual Template Interface for configuration and dynamic application to Virtual Access Interfaces.
Step 3	**mtu** *size* **Example:** Router(config-if)# **mtu 1484**	Sets the MTU size.
Step 4	**ip unnumbered** *interface_type interface_number* **Example:** Router(config-if)#**ip unnumbered vmi 1**	Enable IP processing without an explicit address. Specifies the Virtual Template Interface to use the VMI interface IP address.
Step 5	**no ip redirects** **Example:**	Disables ICMP redirect messages.

	Command or Action	Purpose
	Router(config-if)#**no ip redirects**	
Step 6	**ip tcp adjust-mss** *size* **Example:** Router(config-if)#**ip tcp adjust-mss 1444**	Adjust the mss of transit packets.
Step 7	**load-interval** *value* **Example:** Router(config-if)#**load-interval 30**	Load interval delay.
Step 8	**no peer default ip address** **Example:** Router(config-if)#**no peer default ip address**	Disables peer default ip address.
Step 9	**ipv6 enable** **Example:** Router(config-if)#**ipv6 enable**	Enable IPv6 support under Virtual Template Interface.
Step 10	**ipv6 mtu** *size* **Example:** Router(config-if)#**ipv6 mtu 1484**	Sets the IPv6 MTU size.
Step 11	**ospfv3 1 network** *type* **Example:** Router(config-if)#**ospfv3 1 network manet**	Configure the Virtual Template Interface as MANET OSPF interface type.
Step 12	**ospfv3 1 hello-interval** *value* **Example:** Router(config-if)#**ospfv3 1 hello-interval 10**	Configure the hello interval value for OSPFv3.
Step 13	**ospfv3 1 ipv4 area** *id* **Example:** Router(config-if)#**ospfv3 1 ipv4 area 0**	Enable Virtual Template Interface to participate in OSPFv3 IPv4 routing.
Step 14	**ospfv3 1 ipv6 area** *id* **Example:** Router(config-if)#**ospfv3 1 ipv6 area 0**	Enable Virtual Template Interface to participate in OSPFv3 IPv6 routing.
Step 15	**service-policy input** *policy_map_name* **Example:** Router(config-if)#**service-policy input rar_policer**	Attach the created policy map name to the Virtual Template Interface.

Configuring the Loopback Interface

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**interface loopback** *number* **Example:** Router(config)# **interface loopback 1**	Assigns the loopback interface number.
Step 3	**ip address** *address mask* **Example:** Router(config-if)#**ip address 1.1.1.1 255.255.255.255**	Configuring IPv4 address for the loopback interface.
Step 4	**ipv6 address** *address/prefix* **Example:** Router(config-if)#**ipv6 address 11::11/128**	Configures IPv6 address for the loopback interface.
Step 5	**ipv6 enable** **Example:** Router(config-if)#**ipv6 enable**	Enable IPv6 support under the loopback interface.
Step 6	**ospfv3 1 ipv4 area** *id* **Example:** Router(config-if)#**ospfv3 1 ipv4 area 0**	Enable loopback interface to participate in OSPFv3 routing.
Step 7	**ospfv3 1 ipv6 area** *id* **Example:** Router(config-if)#**ospfv3 1 ipv6 area 0**	Enable loopback interface to participate in OSPFv3 routing.

Configuring the OSPFv3 IPv4 Address Family Process

The OSPFv3 address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, you may have two device processes per interface, but only one process per AF. If the IPv4 AF is used, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface. A single IPv4 or IPv6 OSPFv3 process running multiple instances on the same interface is not supported.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**router ospfv3** *process_id* **Example:** Router(config)# **router ospfv3 1**	Global configuration for OSPFv3.
Step 3	**router-id** *ipv4_address* **Example:** Router(config-router)#**router-id 101.101.101.101**	Router ID for OSPFv3.
Step 4	**address-family ipv4 unicast** **Example:** Router(config-router)#**address-family ipv4 unicast**	Adding address family for IPv4 unicast routing under global OSPFv3 configuration.
Step 5	**redistribute connected metric** *value* **metric-type** *type* **Example:** Router(config-router-af)#**redistribute connected metric 1 metric-type**	Redistribute metrics from external routing protocol.
Step 6	**log-adjacency-changes** **Example:** Router(config-if)#**log-adjacency-changes**	Logs all state changes.

Configuring the OSPFv3 IPv6 Address Family Process

The OSPFv3 address families feature enables both IPv4 and IPv6 unicast traffic to be supported. With this feature, you may have two device processes per interface, but only one process per AF. If the IPv4 AF is used, an IPv4 address must first be configured on the interface, but IPv6 must be enabled on the interface. A single IPv4 or IPv6 OSPFv3 process running multiple instances on the same interface is not supported.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**router ospfv3** *process_id* **Example:**	Global configuration for OSPFv3.

	Command or Action	Purpose
	Router(config)# **router ospfv3 1**	
Step 3	**router-id** *ipv4_address* **Example:** Router(config-router)#**router-id 101.101.101.101**	Router ID for OSPFv3.
Step 4	**address-family ipv6 unicast** **Example:** Router(config-router)#**address-family ipv6 unicast**	Adding address family for IPv6 unicast routing under global OSPFv3 configuration.
Step 5	**redistribute connected metric** *value* **metric-type** *type* **Example:** Router(config-router-af)#**redistribute connected metric 1 metric-type**	Redistribute metrics from external routing protocol.
Step 6	**log-adjacency-changes** **Example:** Router(config-if)#**log-adjacency-changes**	Logs all state changes.

Verifying Virtual Template Interface

This section shows examples of command output to verify your setup.

```
Router# show vtemplate
Virtual access subinterface creation is globally enabled

        Active     Active    Subint  Interface
      Interface Subinterface Capable Type
      --------- ------------ ------- ---------
Vt1           0            1   Yes   Serial

Usage Summary
                              Interface   Subinterface
                              ---------   ------------
Current Serial  in use                2              1
Current Serial  free                  0              1
Current Ether   in use                0              0
Current Ether   free                  0              0
Current Tunnel  in use                0              0
Current Tunnel  free                  0              0
Current VPN  in use                   0              0
Current VPN  free                     0              0
Total                                 2              2

Cumulative created                    3             27
Cumulative freed                      0             26

Base virtual access interfaces: 2
Total create or clone requests: 3
Cancelled create or clone requests: 0
Cumulative create request waiting for sso resources: 0
```

```
Current request queue size: 0
Current free pending: 0
Current recycle pending: 0
Current ordered recycle pending: 0

Maximum request duration: 32 msec
Average request duration: 24 msec
Last request duration: 32 msec

Maximum processing duration: 32 msec
Average processing duration: 24 msec
Last processing duration: 32 msec
Router#
```

Verifying PPPoE Session Details

This section shows examples of command output to verify your setup.

```
Router-1#show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port       VT  VA    StateSID  LocMAC       VA-st      Type

   2     1      000c.296f.c985  Gi0/0/0    1     Vi1.1     PTA    7c31.0e85.1e78
 UP
Router#
```

```
Router-2#show pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port       VT  VA    StateSID  LocMAC       VA-st      Type

   2     1      000c.29a1.ae42  Gi0/0/0    1     Vi1.1     PTA    d478.9b5d.0200
 UP
Router#
```

```
Router-1# show pppoe session packets all
Total PPPoE sessions 1

session id: 1
local MAC address: 7c31.0e85.1e78, remote MAC address: 000c.296f.c985
virtual access interface: Vi1.1, outgoing interface: Gi0/0/0
    67 packets sent, 67 received
    6488 bytes sent, 5908 received

PPPoE Flow Control Stats
 Local Credits: 1953   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 38     PADG Timer index: 89
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 89
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 36  rcvd: 0
 PADC xmit: 127926272  rcvd: 36
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 38, fcn = 65535, bcn = 0
```

```
  PADC rcvd: seq_num = 38, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 61
  PADC xmit: seq_num = 61, fcn = 1952, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0
Router-1#

Router-2# show pppoe session packets all
Total PPPoE sessions 1

session id: 1
local MAC address: d478.9b5d.0200, remote MAC address: 000c.29a1.ae42
virtual access interface: Vi1.1, outgoing interface: Gi0/0/0
    78 packets sent, 75 received
    7408 bytes sent, 6642 received

PPPoE Flow Control Stats
 Local Credits: 1950   Peer Credits: 65535   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 42     PADG Timer index: 96
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 96
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 40  rcvd: 0
 PADC xmit: 127795200  rcvd: 40
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 42, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 42, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 66
  PADC xmit: seq_num = 63, fcn = 1950, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0
Router-2#
```

Verifying VMI Neighbors

This section shows examples of command output to verify your setup.

The Multicast for Virtual Multipoint Interfaces feature enables multicast support for RFC 5578-compliant Radio-Aware Routing (RAR). Multicast is defined as a network group membership spanning the entire network. The virtual multipoint interface (VMI) operates in aggregate mode, which means that all virtual access interfaces created by PPP over Ethernet (PPPoE) sessions are aggregated logically under the configured VMI. Packets sent to the VMI are forwarded to the correct virtual access interface. When a VMI operates in aggregate mode, the interfaces operate in nonbroadcast multiple access (NBMA) mode. Multicast traffic is forwarded only the he NBMA neighbors where a listener for that group is present.

```
Router-1#show vmi neighbor detail
             1 vmi1 Neighbors


vmi1   IPV6 Address=FE80::D678:9BFF:FE5D:200
       IPV6 Global Addr=::
```

```
       IPV4 Address=81.0.0.1, Uptime=00:03:16
       Output pkts=0, Input pkts=0
       METRIC DATA: Total rcvd=2, Avg arrival rate (ms)=10220
          CURRENT: MDR=2000000000000 bps, CDR=1000000000 bps
                   Lat=1 ms, Res=100, RLQ=90, load=1
          MDR      Max=2000000000000 bps, Min=2000000000000 bps, Avg=2000000000000 bps
          CDR      Max=1000000000 bps, Min=1000000000 bps, Avg=1000000000 bps
          Latency  Max=1, Min=1, Avg=1 (ms)
          Resource Max=100%, Min=100%, Avg=100%
          RLQ      Max=90, Min=90, Avg=90
          Load     Max=1%, Min=0%, Avg=0%
       Transport PPPoE, Session ID=2
       INTERFACE STATS:
          VMI Interface=vmi1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access1.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
 Local Credits: 1954   Peer Credits: 65535  Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 198     PADG Timer index: 283
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 283
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 196  rcvd: 0
 PADC xmit: 128057344  rcvd: 196
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 198, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 198, fcn = 1954, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 105
  PADC xmit: seq_num = 109, fcn = 1951, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 1
Router-1#

Router#

Router-2#show vmi neighbor detail
             1 vmi1 Neighbors


vmi1   IPV6 Address=FE80::7E31:EFF:FE85:1E78
       IPV6 Global Addr=::
       IPV4 Address=71.0.0.1, Uptime=00:01:50
       Output pkts=0, Input pkts=0
       No Session Metrics have been received for this neighbor.
       Transport PPPoE, Session ID=1
       INTERFACE STATS:
          VMI Interface=vmi1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          V-Access intf=Virtual-Access1.1,
             Input qcount=0, drops=0, Output qcount=0, drops=0
          Physical intf=GigabitEthernet0/0/0,
             Input qcount=0, drops=0, Output qcount=0, drops=0
```

```
PPPoE Flow Control Stats
 Local Credits: 1953   Peer Credits: 65533   Local Scaling Value 65534 bytes
 Credit Grant Threshold: 28000    Max Credits per grant: 65535
 Credit Starved Packets: 0
 PADG xmit Seq Num: 112     PADG Timer index: 168
 PADG last rcvd Seq Num: 0
 PADG last nonzero Seq Num: 168
 PADG last nonzero rcvd amount: 0
 PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
 PADG xmit: 110  rcvd: 0
 PADC xmit: 127991808  rcvd: 110
 In-band credit pkt xmit: 0 rcvd: 0
 Last credit packet snapshot
  PADG xmit: seq_num = 112, fcn = 65535, bcn = 0
  PADC rcvd: seq_num = 112, fcn = 1953, bcn = 65535
  PADG rcvd: seq_num = 0, fcn = 0, bcn = 70
  PADC xmit: seq_num = 71, fcn = 1952, bcn = 0
  In-band credit pkt xmit: fcn = 0, bcn = 0
  In-band credit pkt rcvd: fcn = 0, bcn = 0
    ==== PADQ Statistics ====
     PADQ xmit: 0  rcvd: 0

Router-2#
```

Verifying OSPF Neighbor

This section shows examples of command output to verify your setup.

```
Router-1#sh ospfv3 neighbor | i FULL
102.102.102.102   0   FULL/  -        00:00:33    24    Virtual-Access1.1
102.102.102.102   0   FULL/  -        00:00:34    24              Virtual-Access1.1
Router-1

Router-2# sh ospfv3 neighbor | i FULL
sh ospfv3 neighbor | i FULL
101.101.101.101   0   FULL/  -        00:00:33    23            Virtual-Access1.1
101.101.101.101   0   FULL/  -        00:00:33    23              Virtual-Access1.1
Router-2#
```

CHAPTER 294

RAR PPPoE IPv6 Multicast

From Cisco IOS XE 17.11.1a, IPv6 multicast configuration is available for PPPoE-based RAR sessions.

Prerequisites

For IPv6 multicast over PPPoE to function properly, the following must be configured:

- PPPoE (Virtual-template, VMI and physical interface)
- IPv6 unicast and multicast routing
- IPv6 PIM BSR
- IPv6 MLD

Configuring VMI interface and Enabling Multicast Support

Used in router-to-radio communications, the Virtual Multipoint Interface (VMI) interface provides services that map outgoing packets to the appropriate Point-to-Point Protocol over Ethernet (PPPoE) sessions based on the next-hop forwarding address for that packet. The VMI interface also provides a broadcast service that emulates a set of point-to-point connections as a point-to-multipoint interface with broadcast ability. When a packet with a multicast address is forwarded through the VMI interface, VMI replicates the packet and unicasts it to each of its neighbors.

Note VMI will map outgoing packets to the appropriate PPPoE sessions. It will use the next-hop forwarding address from each outgoing packet to perform this mapping.

Note VMI is required to have IP addresses assigned for VMI to work even though it will be shown as down/down while in bypass mode. The IPv4 address configured will not be advertised or used. Instead, the IPv4 address on the virtual-template will be used.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**interface vmi** *number* **Example:** Router(config)# **interface vmi 1**	Creates a VMI and enters interface configuration mode.
Step 3	**ipv6 enable** **Example:** Router(config-if)#**ipv6 enable**	Enable IPv6 support under VMI interface. OSPFv3 IPv4 needs to have IPv6 support enabled on the interface level.
Step 4	**ip address** *address mask* **Example:** Router(config-if)#**ip address 71.0.0.1 255.255.255.0**	Configures IPv4 address under VMI interface.
Step 5	**ipv6 address** *address/prefix* **Example:** Router(config-if)#**ipv6 address 71::71/64**	Configures IPv6 address under VMI interface.
Step 6	**physical-interface** *interface* **Example:** Router(config-if)#**physical-interface gigabitEthernet {0/0/0 or 0/0/1 or 0/0/0.1}**	Binds the physical interface (the interface connected to the radio client) to the VMI interface, for packet flow.
Step 7	**mode** *name* **Example:** Router(config-if)#**mode bypass**	Do not aggregate Virtual Access Interfaces under VMI. (Must be configured as bypass).

Configuring IPv6 PIM Bootstrap Router (BSR)

The bootstrap router (BSR) protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes

unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**ipv6 pim bsr candidate bsr** *loopback_ipv6_address* **Example:** Router(config)# **ipv6 pim bsr candidate bsr 11::11**	Configures a device to be a candidate BSR. **Note** Must use the loopback ipv6 address configured earlier.
Step 3	**ipv6 pim bsr candidate rp** *loopback_ipv6_address* **Example:** Router(config-router)#**ipv6 pim bsr candidate rp 11::11**	Configures a device to be a candidate Rendezvous Point (RP). **Note** Must use the loopback ipv6 address configured earlier.

Configuring IPv6 Multicast Group

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries--receivers can be located anywhere on the Internet or in any private network.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface_name* **Example:** Router# **interface Vlan 10**	Creates and enters interface configuration mode.
Step 3	**ipv6 mld join-group** *multicast_group_address* **Example:** Router(config-if)#**ipv6 mld join-group FF06:6::1**	Configure IPv6 mld join group under interface.

Verifying BSR Election

This section shows examples of command output to verify your setup.

```
Router-1#show ipv6 pim bsr election
PIMv2 BSR information

BSR Election Information
  Scope Range List: ff00::/8
  This system is the Bootstrap Router (BSR)
     BSR Address: 61::61
     Uptime: 00:05:20, BSR Priority: 0, Hash mask length: 126
     RPF: FE80::7E31:EFF:FE85:1E78,Loopback0
     BS Timer: 00:00:41
  This system is candidate BSR
       Candidate BSR address: 61::61, priority: 0, hash mask length: 126
Router-1#

Router-2#show ipv6 pim bsr election
PIMv2 BSR information

BSR Election Information
  Scope Range List: ff00::/8
     BSR Address: 61::61
     Uptime: 00:01:22, BSR Priority: 0, Hash mask length: 126
     RPF: FE80::7E31:EFF:FE85:1E78,Virtual-Access1.1
     BS Timer: 00:01:47
  This system is candidate BSR
       Candidate BSR address: 41::41, priority: 0, hash mask length: 126
Router-2#
```

Verifying IPv6 Multicast Configuration

This section shows examples of IPv6 multicast ocnfiguration.

Note "*" in the source address indicates MLD join request entry.

```
Router-1#show ipv6 mroute
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(A000::2, FF06:6::1), 00:01:45/00:01:44, flags: SFT
  Incoming interface: Vlan30
  RPF nbr: A000::2
```

```
  Immediate Outgoing interface list:
    Virtual-Access1.1, Forward, 00:01:45/00:02:48

Router-1#

Router-2# show ipv6 mroute
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF06:6::1), 00:04:33/00:03:29, RP 41::41, flags: SCL
  Incoming interface: Tunnel2
  RPF nbr: 41::41
  Immediate Outgoing interface list:
    Vlan30, Forward, 00:04:33/00:03:29

(A000::2, FF06:6::1), 00:01:41/00:02:43, flags: ST
  Incoming interface: Virtual-Access1.1
  RPF nbr: FE80::7E31:EFF:FE85:1E78
  Immediate Outgoing interface list:
    Vlan30, Forward, 00:01:41/00:02:49
Router-2
```

Sample Running Configuration

The following output shows a sample configuration for IPv6 multicast over RAR PPPoE.

```
Router-1#show run
Building configuration...

Current configuration : 7911 bytes
!
! Last configuration change at 13:04:28 UTC Wed Jun 8 2022
!
version 17.9
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform hardware throughput level 250M
platform punt-keepalive disable-kernel-core
!
hostname Router-1
!
boot-start-marker
boot system flash bootflash:c6300-universalk9.17.09.01.SSA.bin
boot-end-marker
!
!
no logging console
no aaa new-model
!
```

```
!
login on-success log
ipv6 unicast-routing
ipv6 multicast-routing
!
!
subscriber templating
subscriber authorization enable
!
!
multilink bundle-name authenticated
!
!
crypto pki trustpoint TP-self-signed-4073554590
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-4073554590
 revocation-check none
 rsakeypair TP-self-signed-4073554590
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
!
crypto pki certificate chain TP-self-signed-4073554590
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 34303733 35353435 3930301E 170D3232 30363038 31313137
  31345A17 0D333230 36303731 31313731 345A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34 30373335
  35343539 30308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100C54E 860F73AC F0A9EBC7 2C6D1204 49099324 85989550 32CA9B91
  3B3A2492 AEA1D550 0CD787DE 09F6B64F 3F01C578 3EFB3995 E448904A 957EAFD9
  B82EF201 3A28BE26 6B1615DA 35B35BEA D4B7B20F 2D2A3EC2 C1F52281 349E88E5
  F2BCBA37 CA72D461 97D1E817 7493ED38 7C7C1035 F7231D4E F59FADF9 EE0EE5EE
  1FD73691 E93EBE8C 262DC8B1 0FA25BF0 C2F65BF9 C57A406A 9F9CF3D3 3E6888C3
  D6B533AB 0DA71037 6C94A385 CECA4DD9 A037C344 5B761E6D F3B8D47B 4093BED8
  E497D649 63436773 7BE5A718 331C7F08 31071542 03AE588C 08605CF7 CC7C7D6F
  967759FC E2A943FF 8AD70094 825AAD4B DC66FEC8 5B7F2CDA 24F148E1 51AF106E
  FE212C21 651B0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
  301F0603 551D2304 18301680 14F49EA0 C5E78BB1 F0EA55AD 4C580FB7 BF1AE35C
  50301D06 03551D0E 04160414 F49EA0C5 E78BB1F0 EA55AD4C 580FB7BF 1AE35C50
  300D0609 2A864886 F70D0101 05050003 82010100 2083EBC1 E760806B 12F0ACEF
  8FDE2E11 E5B88B62 B85B5835 AB1D2471 EA6FCC2D B28F5EE2 1969C233 DBB8C435
  D7BAD49C 7781E485 97D5B5D0 DD05A0EE 5352535F 1657BE78 64E6BBA8 B627618E
  49041DFB 4FE4D16C FA6857EF E6EEDFBC 2E25AF9E 852EDF71 3B65E55F 62AB1B1E
  8B842F51 DAD55DB5 8A5BF87B 91F540D2 02E8576E 5D4550EA E7FCB6D7 6AD0E92A
  EEAF7544 01C4095A BB02DB3A D45D73EB 971974FF B5DED058 F2F3A0E1 23BD3441
  899CDFC2 A3B36E7D E72D4BDB 480B8347 C26D6AB9 E7E5A140 B20B7B1D 7AC24A2C
  A69124B3 49BF18AF C99EF2A5 C4F484CE 9E2A70C1 D1EA4250 6E0D858E BBCFC6C1
  4FF6E0BC EB190067 E86EC80D 5D149D6F 462CA857
   quit
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
```

```
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
   quit
!
!
no license feature hseck9
license udi pid ESR-6300-CON-K9 sn FOC234304H3
license boot level network-advantage
memory free low-watermark processor 45135
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
redundancy
 mode none
!
!
vlan internal allocation policy ascending
!
policy-map type service pppoe_rar
 pppoe service manet_radio
!
!
bba-group pppoe rar_group_1
 virtual-template 1
 service profile pppoe_rar
!
!
interface Loopback0
 ip address 61.0.0.1 255.255.255.255
 ipv6 address 61::61/128
 ipv6 enable
 ospfv3 1 ipv6 area 0
 ospfv3 1 ipv4 area 0
!
interface GigabitEthernet0/0/0
 media-type rj45
 negotiation auto
 ipv6 enable
 pppoe enable group rar_group_1
!
interface GigabitEthernet0/0/1
```

```
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/0
 switchport access vlan 30
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface Virtual-Template1
 mtu 1484
 ip unnumbered vmi1
 no ip redirects
 ip tcp adjust-mss 1444
 load-interval 30
 no peer default ip address
 ipv6 enable
 ipv6 mtu 1484
 ospfv3 1 network manet
 ospfv3 1 hello-interval 10
 ospfv3 1 ipv4 area 0
 ospfv3 1 ipv6 area 0
!
interface Vlan1
 no ip address
!
interface Vlan30
 ip address 192.168.10.1 255.255.255.0
 ipv6 address A000::1/64
 ipv6 enable
 ospfv3 1 ipv4 area 0
 ospfv3 1 ipv6 area 0
!
interface Async0/2/0
 no ip address
 encapsulation scada
!
interface vmi1
 ip address 71.0.0.1 255.255.255.0
 ipv6 address FE80::7E31:EFF:FE85:1E78 link-local
 ipv6 address 71::71/64
 ipv6 enable
 physical-interface GigabitEthernet0/0/0
 mode bypass
!
router ospfv3 1
 router-id 101.101.101.101
 !
 address-family ipv4 unicast
  redistribute connected metric 1 metric-type 1
  log-adjacency-changes
 exit-address-family
```

```
 !
 address-family ipv6 unicast
  redistribute connected metric-type 1
  log-adjacency-changes
 exit-address-family
!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!
ipv6 pim bsr candidate bsr 61::61
ipv6 pim bsr candidate rp 61::61
!
!
control-plane
!
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 stopbits 1
line 0/0/0
line 0/2/0
line vty 0 4
 login
 transport input ssh
line vty 5 14
 login
 transport input ssh
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email
 address to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
!

end
```

Debug Commands

This section shows debug commands for PPPoE, VMI, Virtual Template, and Subscriber Service.

PPPoE

Command or Action	Purpose
Router# **debug pppoe data**	PPPoE data packets
Router# **debug pppoe errors**	PPPoE protocol errors
Router# **debug pppoe events**	PPPoE protocol events
Router# **debug pppoe packets**	PPPoE control packets

VMI

Command or Action	Purpose
Router# **debug vmi bma**	Display VMI bma debug
Router# **debug vmi error**	Display internal VMI anomalies detected
Router# **debug vmi multicast**	Display VMI multicast packets
Router# **debug vmi neighbor**	Display VMI neighbor transaction debugging
Router# **debug vmi packet**	Display VMI packet trace
Router# **debug vmi pppoe**	Display VMI PPPoE packet/activity debug
Router# **debug vmi registries**	Display VMI registry calls

Virtual Template

Command or Action	Purpose
Router# **debug vtemplate cloning**	Virtual Template cloning information
Router# **debug vtemplate error**	Virtual Template errors
Router# **debug vtemplate event**	Virtual Template events
Router# **debug vtemplate subinterface**	Virtual Template subinterface command

Subscriber Service

Command or Action	Purpose
Router# **debug sss errors**	Subscriber Service Switch Manager errors
Router# **debug ssss event**	Subscriber Service Switch Manager events