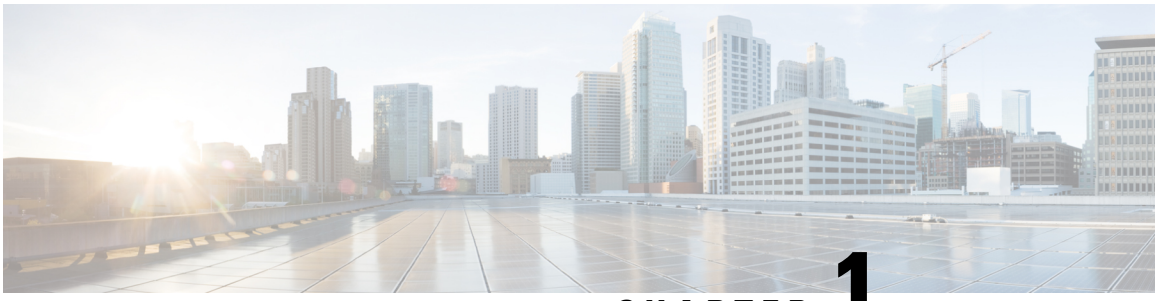
IP Routing: Protocol-Independent Configuration Guide, Cisco IOS XE Gibraltar 16.12.x

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



CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

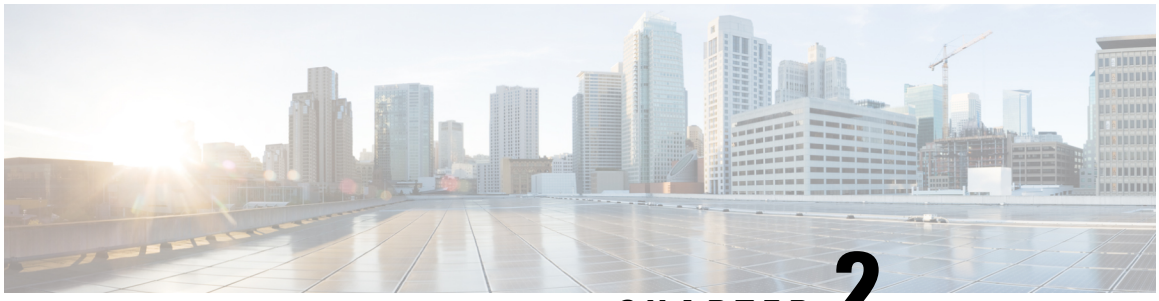
Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- 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 Marketplace.
- 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.



CHAPTER 2

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.

Note A packet with an E-class source address (240.0.0.0/4) gets dropped on Cisco ASR 1000 Series Aggregation Services Routers, although RFC 1812 (Requirements for IP Version 4 Routers) defines this behavior only for destination addresses and not specifically for source addresses.

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.

Command or Action	Purpose
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.

Command or Action	Purpose
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.
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**	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* **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:** `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

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.

	Command or Action	Purpose
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.
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.
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
```

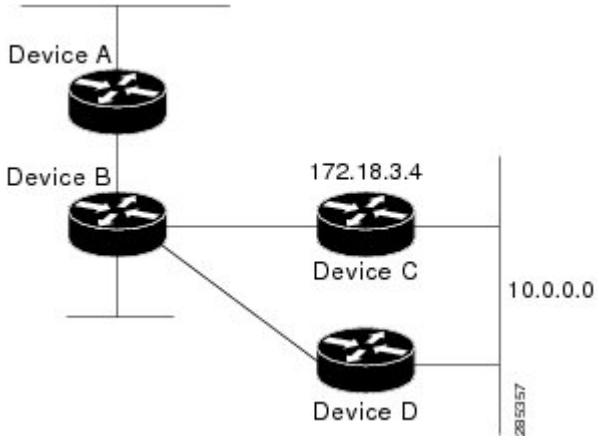
```
Device(config-if)# ip address 172.16.255.5 255.255.255.252
Device(config-if)# ! 2 bits of address space reserved for point-to-point serial interfaces
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 172.16.0.0
Device(config-router)# ! Specifies the network directly 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.

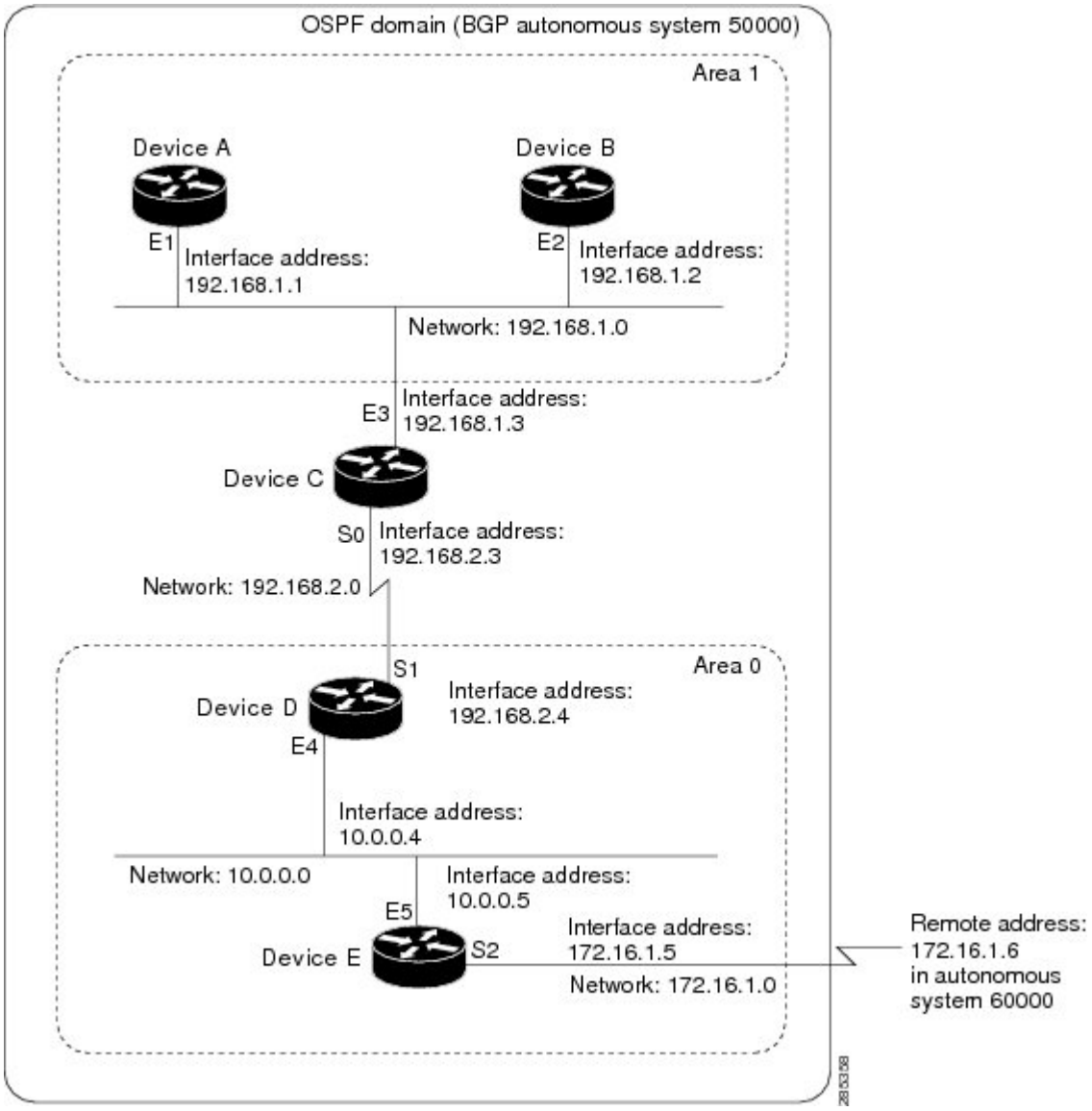
The second **network** command is evaluated next. For Area 2, the same process is then applied to all interfaces (except Gigabit Ethernet interface 0/0/0). Assume that a match is determined for Gigabit Ethernet interface 1/0/0. OSPF is then enabled for that interface, and Gigabit Ethernet 1/0/0 is attached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

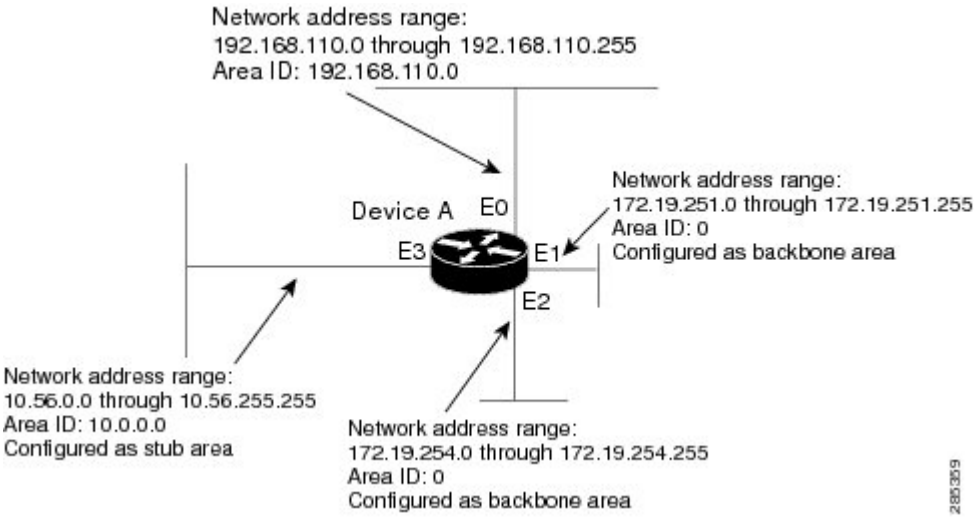
Example: Complex OSPF Configuration

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

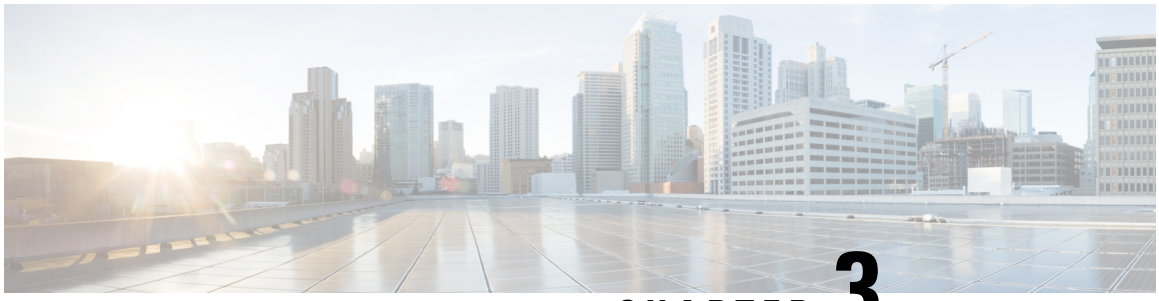
Feature Information for Basic IP Routing

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

Feature Name	Releases	Feature Information
IP Routing		The IP Routing feature introduced basic IP routing features that are documented throughout this module and also in other IP Routing Protocol modules.



CHAPTER 3

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

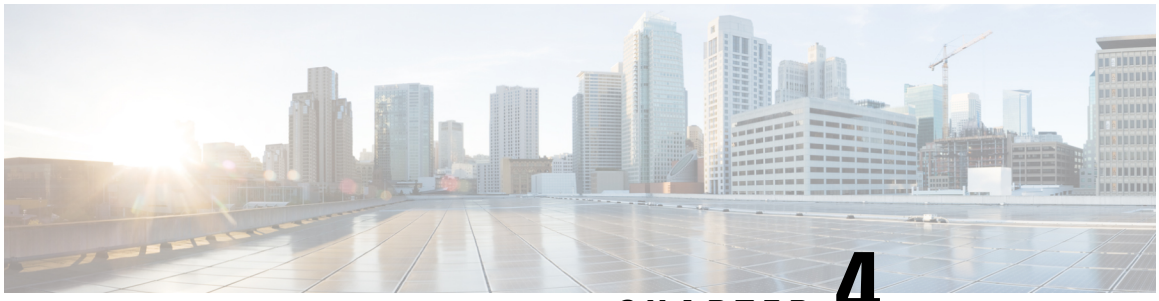
Feature Information for IPv6 Routing: Static Routing

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

Feature Name	Releases	Feature Information
IPv6 Routing: Static Routing	12.0(22)S 12.2(2)T 12.2(14)S 12.2(17a)SX1 12.2(25)SG 12.2(28)SB 12.2(33)SRA Cisco IOS XE Release 2.1	Static routes are manually configured and define an explicit path between two networking devices. The following commands were introduced or modified: **ipv6 route**, **ipv6 route static resolve default**, **show ipv6 route**, **show ipv6 static**.



CHAPTER 4

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.

The following restrictions apply to ASR 903 series Aggregation Services Routers:

- To enable LFA FRR on Cisco ASR 903 series Aggregation Services Routers, you must enable the **mpls ldp explicit-null** command; the **implicit-null** keyword is not supported.
- The ASR 903 supports up to 4000 LFA FRR routes.
- LFA FRR is not supported with equal cost multipath (ECMP).
- Remote LFA tunnels are not High Availability aware; hence, they are Stateful Switchover (SSO) coexistent but not SSO compliant.
- Fast Reroute triggered by Bidirectional Forwarding (BFD) is not supported. Do not configure BFD on any interface that is part of a LFA FRR topology.

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.

	Command or Action	Purpose
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:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
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.

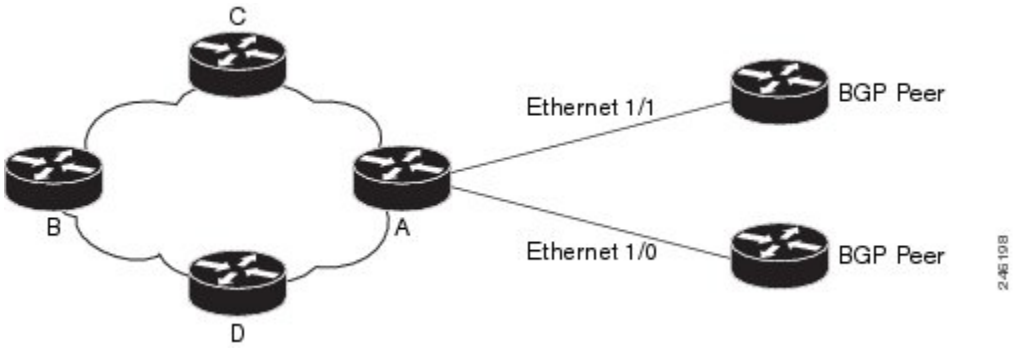
	Command or Action	Purpose
Step 13	**router isis** *area-tag* **Example:** `Device(config)# router isis ipfrr`	Enables the IS-IS routing protocol, specifies an IS-IS process, and enters router configuration mode.
Step 14	**net** *net* **Example:** `Device(config-router)# net 49.0001.0101.2800.0001.00`	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.
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 4: 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
```

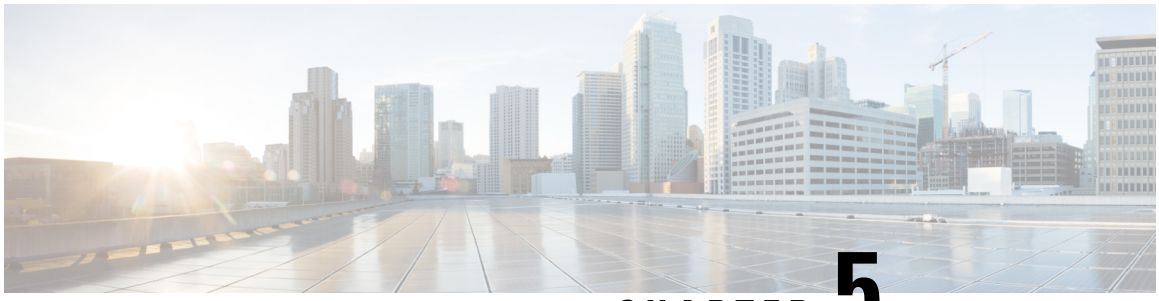
Feature Information for Configuring IPv4 Loop-Free Alternate Fast Reroute

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

Feature Name	Releases	Feature Information
IPv4 Loop-Free Alternate Fast Reroute		When a link or router fails, distributed routing algorithms compute new routes that take into account the change. The time taken for computation is called the routing transition. Until the transition is complete and all routers are converged on a common view of the network, 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 focuses on the precomputation of repair paths. The repair path computation is done by the IS-IS routing protocol and the results (the repair paths) are sent to the RIB. The repair path installation is done by Cisco Express Forwarding. In Cisco IOS XE Release 3.6S, this feature was introduced in ASR 903 Series Aggregation Services Routers. The following commands were introduced or modified: **debug isis fast-reroute**, **fast-reroute load-sharing disable**, **fast-reroute per-prefix**, **fast-reroute tie-break**, **show isis fast-reroute**.



CHAPTER 5

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 20,000 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*
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 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 5: Feature Information for IP Event Dampening

Feature Name	Releases	Feature Information
IP Event Dampening	Cisco IOS XE Release 2.1	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. This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced by this feature: **dampening**, **debug dampening**, **show dampening interface**, **show interface 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.

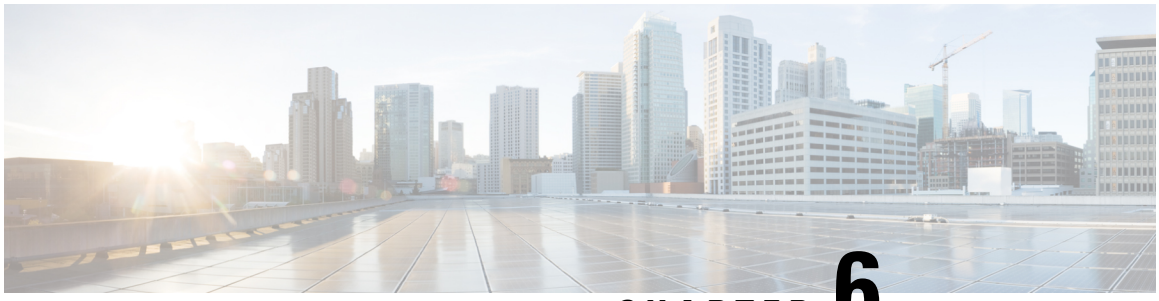
maximum suppress time --The maximum amount of time the interface can stay suppressed at the time a penalty is assigned.

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 6

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.

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 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:**	Enables policy routing and enters route-map configuration mode.

	Command or Action	Purpose
	`Router(config)# route-map abccomp`	

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:
	• **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`
		Sets a recursive next-hop IPv4/IPv6 address.
		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

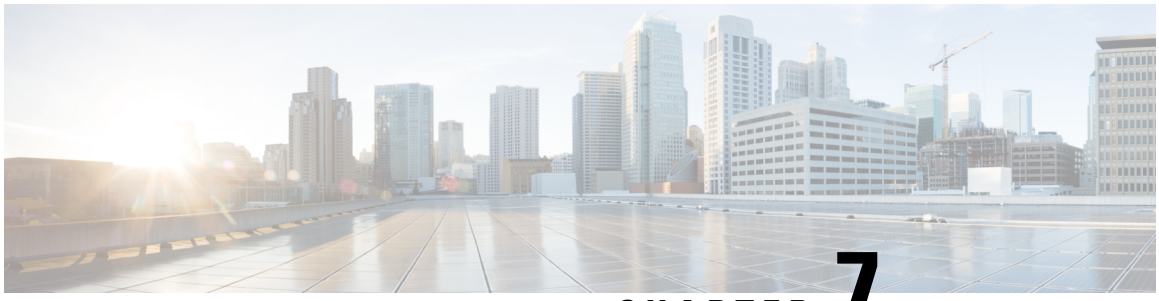
Feature Information for PBR Recursive Next Hop

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 6: Feature Information for PBR Recursive Next Hop

Feature Name	Releases	Feature Information



CHAPTER 7

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

The tasks in this section are divided according to the Cisco IOS release that you are running because Cisco IOS Release 12.3(14)T introduced new syntax for IP Service Level Agreements (SLAs). To use this feature, you must be running Cisco IOS Release 12.3(4)T, 12.2(25)S, or a later release. This section contains the following tasks:

Cisco IOS Release 12.3(11)T 12.2(25)S and Earlier

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.

Before you begin

This task requires the networking device to be running Cisco IOS Release 12.3(11)T, 12.2(25)S, or prior releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **rtr** *operation-number*
4. **type echo protocol** *protocol-type target* [**source-ipaddr** *ip-address*]
5. **exit**
6. **rtr schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*]
7. **track** *object-number* **rtr** *entry-number* [**reachability**]
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**

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	**rtr** *operation-number* **Example:** `Router(config)# rtr 1`	Enters SAA RTR configuration mode and configures an SAA operation.
Step 4	**type echo protocol** *protocol-type target* [**source-ipaddr** *ip-address*] **Example:** `Router(config-rtr)# type echo protocol ipicmpecho 10.1.1.10`	Configures an SAA end-to-end echo response time probe operation.
Step 5	**exit** **Example:** `Router(config-rtr)# exit`	Exits SAA RTR configuration mode and returns the router to global configuration mode.
Step 6	**rtr schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh* **:** *mm*[**:** *ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh* **:** *mm* **:** *ss*}] [**ageout** *seconds*] **Example:** `Router(config)# rtr schedule 1 life forever start-time now`	Configures the time parameters for the SAA operation.
Step 7	**track** *object-number* **rtr** *entry-number* [**reachability**] **Example:** `Router(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
	`Router(config-track)# delay up 60 down 30`	
Step 9	**exit** **Example:** `Router(config-track)# exit`	Exits tracking configuration mode and returns the router to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface ethernet 0`	Specifies an interface type and number and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.1.1.11 255.0.0.0`	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.
Step 12	**ip policy route-map** *map-tag* **Example:** `Router(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:** `Router(config-if)# exit`	Exits interface configuration mode and returns the router to global configuration mode.
Step 14	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map alpha`	Specifies a route map and enters route-map configuration mode.
Step 15	**set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*] **Example:** `Router(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.
Step 16	**end** **Example:** `Router(config-route-map)# end`	Exits route-map configuration mode and returns the router to privileged EXEC mode.

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.

	Command or Action	Purpose
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:** `Device(config-track)# delay up 60 down 30`	(Optional) Specifies a period of time, in seconds, to delay communicating state changes of a tracked object.
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:**	Specifies a primary or secondary IP address for an interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 192.168.1.1 255.255.255.0`	• 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.
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

Cisco IOS Release 12.3(11)T 12.2(25)S and Earlier

In the following example, object tracking is configured for PBR on routers that are running Cisco IOS Release 12.3(11)T, 12.2(25)S, or earlier releases.

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 Response Time Reporters (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.
rtr 1
 type echo protocol ipicmpecho 10.1.1.1
rtr schedule 1 start-time now life forever
!
rtr 2
 type echo protocol ipicmpecho 10.2.2.2
rtr 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
```

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**

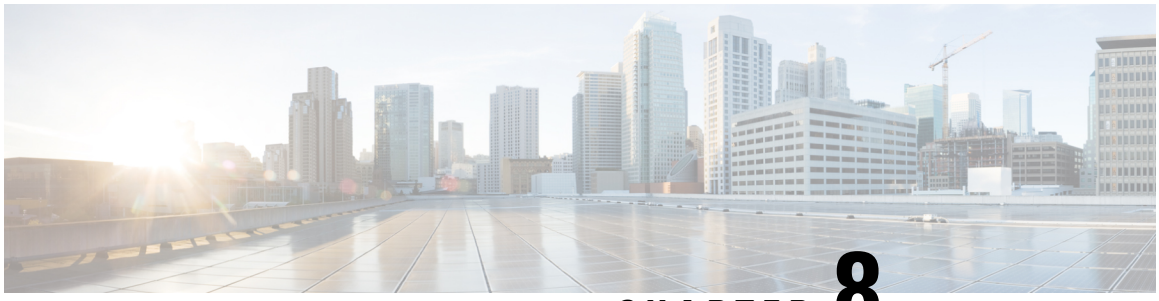
Feature Information for PBR Support for Multiple Tracking Options

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 7: Feature Information for PBR Support for Multiple Tracking Options

Feature Name	Releases	Feature Information
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. The following commands were introduced or modified by this feature: **set ip next-hop verify-availability**.



CHAPTER 8

PBR Match Track Object

The PBR Match Track Object feature enables a device to track the stub object during 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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for PBR Match Track Object

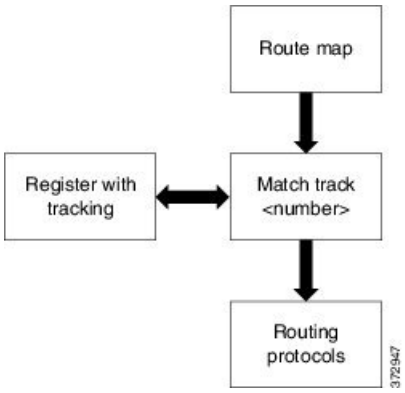
- You can use only one match track variable at a time in a route map sequence.
- You must remove the existing match track object configuration before configuring another match track object. The match track object is unregistered from the tracking component when you remove the match track object number 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

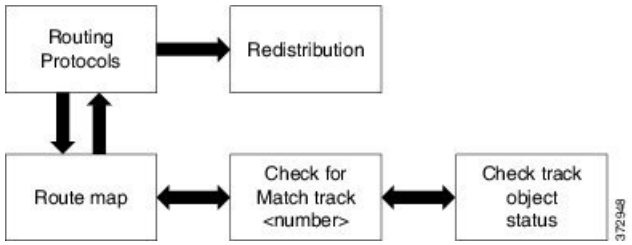
You refer to the stub object that you track as the match track object. The device checks for the existence of the match track object and issues an error message if there is none. Then registration with the tracking component is done to track this object. The device issues an error in case the registration fails.

Figure 5: 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 6: 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

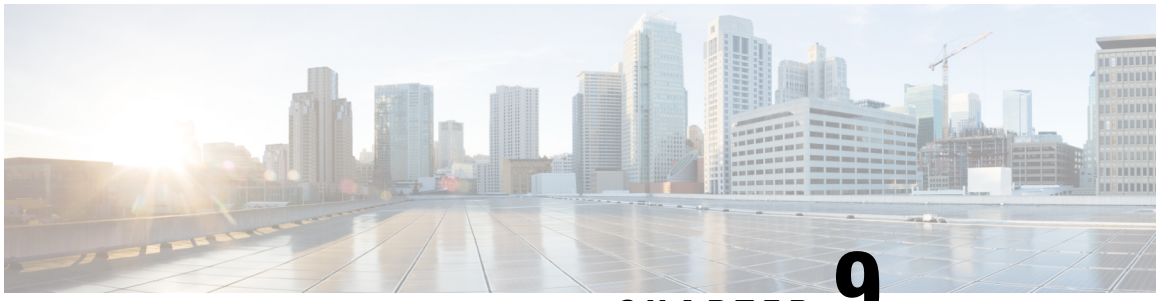
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, 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.

Feature Name	Releases	Feature Information
PBR Match Track Object		The PBR Match Track Object feature enables a device to track the stub object during Policy Based Routing. The following command was introduced: **match track** *track-object-number*



CHAPTER 9

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.

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 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. IPv6 PBR is supported on Cisco ASR 1000 Series platform.

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**	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	**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:
		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

	`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`
	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:** `Device> 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 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

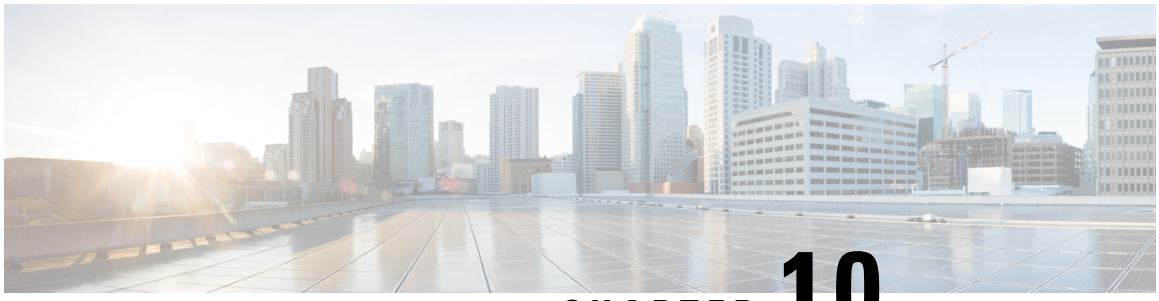
Feature Information for IPv6 Policy-Based Routing

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 8: Feature Information for IPv6 Policy-Based Routing

Feature Name	Releases	Feature Information
IPv6 Policy-Based Routing		Policy-based routing for IPv6 allows a user to manually configure how received packets should be routed. The following commands were introduced or modified: **debug fm ipv6 pbr**, **debug ipv6 policy**, **ipv6 local policy route-map**, **ipv6 policy route-map**, **match ipv6 address**, **match length**, **route-map**, **set default interface**, **set interface**, **set ipv6 default next-hop**, **set ipv6 next-hop (PBR)**, **set ipv6 precedence**, **set vrf**, **show fm ipv6 pbr all**, **show fm ipv6 pbr interface**, **show ipv6 policy**, and **show route-map**.



CHAPTER 10

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.

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 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.

RestrictionsforMulti-VRFSelectionUsingPolicy-BasedRouting

- 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. This is applicable only for Cisco ASR 1000 platform.
- 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.
- **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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
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.

	Command or Action	Purpose
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:** `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*] [**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`
		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.

	Command or Action	Purpose

	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`	

Step 6	Do one of the following:
	• **match ip address** {*acl-number* [*acl-name* | *acl-number*]}
	• **match length** *minimum-lengthmaximum-length*
	Example:
	`Device(config-route-map)# match ip address 1 or`
	Example:
	`Device(config-route-map)# match length 3 200`
		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.
		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.
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 9: Feature Information for Multi-VRF Selection Using Policy-Based Routing

Feature Name	Releases	Feature Information
Multi-VRF Selection Using Policy-Based Routing (PBR)	12.2(33)SRB1 12.2(33)SXH1 12.4(24)T Cisco IOS XE Release 2.2	The Multi-VRF Selection Using Policy-Based Routing (PBR) feature allows a specified interface on a provider edge (PE) router to route packets to Virtual Private Networks (VPNs) based on packet length or match criteria defined in an IP access list. This feature and the MPLS VPN VRF Selection Based on Source IP Address feature can be configured together on the same interface In Cisco IOS Release 12.2(33)SRB1, this feature was introduced. In Cisco IOS Release 12.2(33)SXH1, support was added. In Cisco IOS Release 12.4(24)T, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were modified: **set ip global next-hop** and **set ip vrf next-hop**.
IPv6 VRF-Aware PBR Next-hop Enhancement	15.2(2)S Cisco IOS XE Release 3.6S	In Cisco IOS Release 15.2(2)S, this feature was introduced. In Cisco IOS XE Release 3.6S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced: **set ipv6 default next-hop**, **set ipv6 next-hop (PBR)**

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.

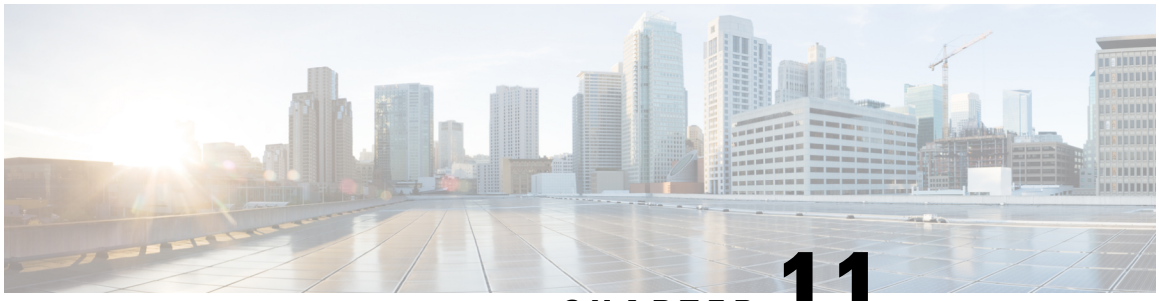
PBR—policy-based routing. PBR allows a user to manually configure how received packets should be routed.

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 11

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.

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 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.

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.

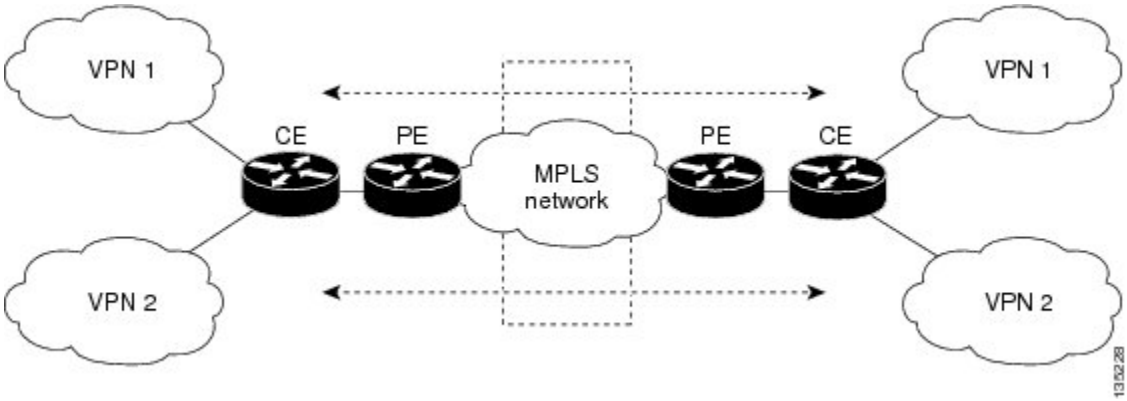
The Multi-VRF Support feature works as follows:

- 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 7: 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.

The following are the supported flavors of multicast over VRF on Cisco ASR 920 RSP2 module:

- Multicast with multi-VRF (MPLS VPN/MLDP)
- Multicast with GRE tunnel (MVPN GRE)
- Multicast with VRF-lite

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.

	Command or Action	Purpose
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:** `Device(config-vrf)# exit`	Returns to global configuration mode.
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.

	Command or Action	Purpose
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:** `Device(config-router-af)# neighbor 10.0.0.3 remote-as 100`	Informs this device's BGP neighbor table of the neighbor's address (or peer group name) and the neighbor's autonomous system number.
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:** `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	**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:**	Enters interface configuration mode for the interface to be used for the BGP session. The interface can be a routed port or an .

	Command or Action	Purpose
	`Device(config)# interface`	
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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
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:** `Device(config-router)# log-adjacency-changes`	(Optional) Logs changes in the adjacency state. This is the default state.
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**	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 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

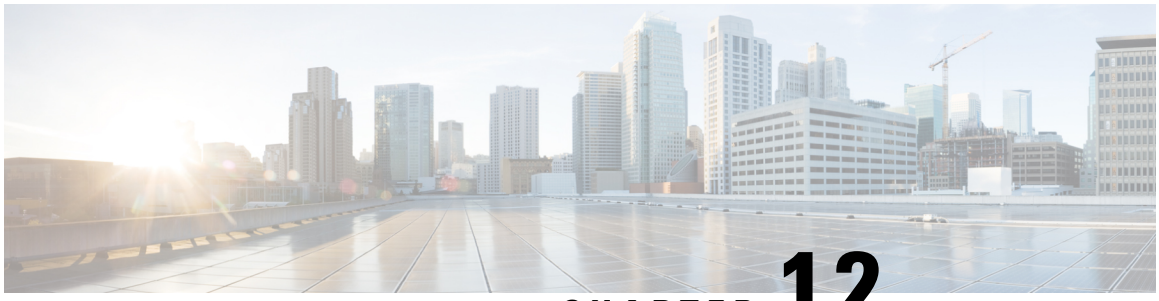
Feature Information for Multi-VRF Support

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 10: Feature Information for Multi-VRF Support

Feature Name	Releases	Feature Information
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 CE device.



CHAPTER 12

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.

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. • *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.

	Command or Action	Purpose
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

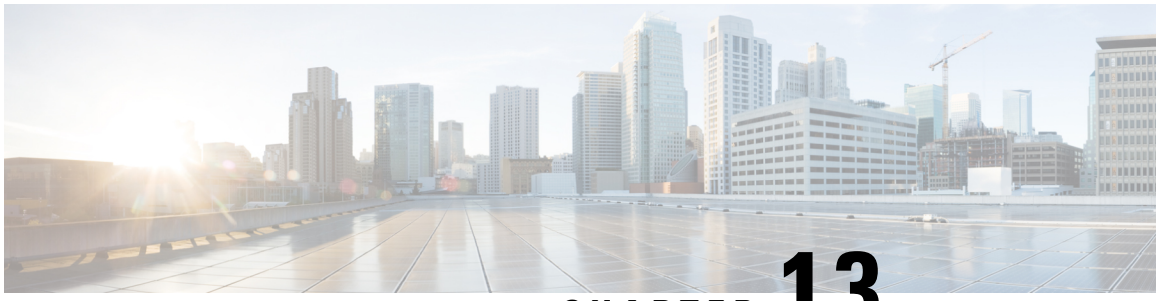
Feature Information for Default Passive Interfaces

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 11: Feature Information for Default Passive Interfaces

Feature Name	Releases	Feature Information
Default Passive Interfaces		In ISP and large enterprise networks, many of the distribution devices have more than 200 interfaces. Obtaining routing information from these interfaces required configuration of the routing protocol on all interfaces and manual configuration of the **passive-interface** command on the interfaces where adjacency was not desired. The Default Passive Interface feature simplifies the configuration of distribution devices by allowing all interfaces to be set as passive by default using a single **passive-interface default** command, and then by configuring individual interfaces where adjacencies are desired using the **no passive-interface** command.



CHAPTER 13

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 12: 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

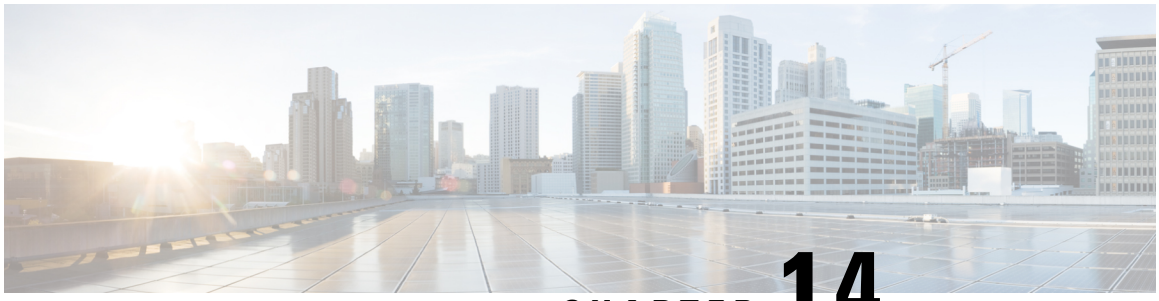
Feature Information for Policy-Based Routing

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 13: Feature Information for Policy-Based Routing

Feature Name	Releases	Feature Information
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 introduces a more flexible mechanism for routing packets than destination routing. The following command was introduced or modified: **ip policy route-map**.



CHAPTER 14

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.

Information About Enhanced Policy-Based Routing and Site Manager

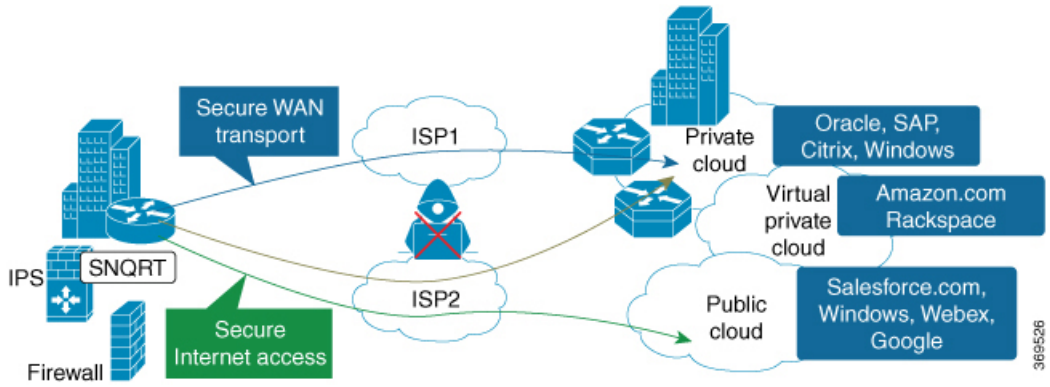
About Enhanced Policy-Based Routing and Site Manager

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 8: 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.
- Internet Edge with Multihoming - On the internet edge with multiple ISP links, you can define a policy to forward specific traffic to one ISP or load balance among the existing ISP links.
- 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.

Configure Enhanced Policy-Based and Site Manager

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

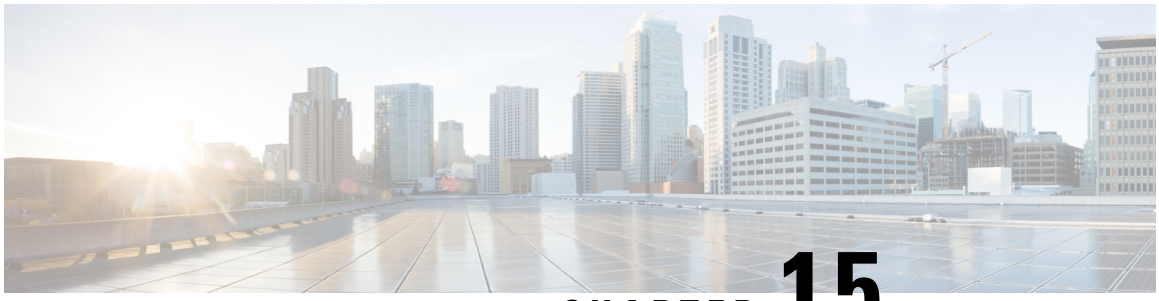
```
enable
configure terminal
    site-manager default
      vrf default
        border
         source-interface loopback0
         master 192.168.3.22
        exit
       exit

  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
```

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 14: Feature Information for Overview of Cisco TrustSec

Feature Name	Releases	Feature Information
ePBR-Application-Based Routing	Cisco IOS XE Gibraltar 16.11.1	As network-based applications start being hosted on private or public cloud, network appliances need to 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, while also ensuring application performance.



CHAPTER 15

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.

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 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

Related Documents

Related Topic	Document Title
Cisco CSR 1000V VxLAN Support	Cisco CSR 1000V VxLAN Support
Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide	Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide
IP Routing: Protocol-Independent Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)	IP Routing: Protocol-Independent Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)
IP Routing: Protocol-Independent Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 920 Series)	IP Routing: Protocol-Independent Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 920 Series)

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

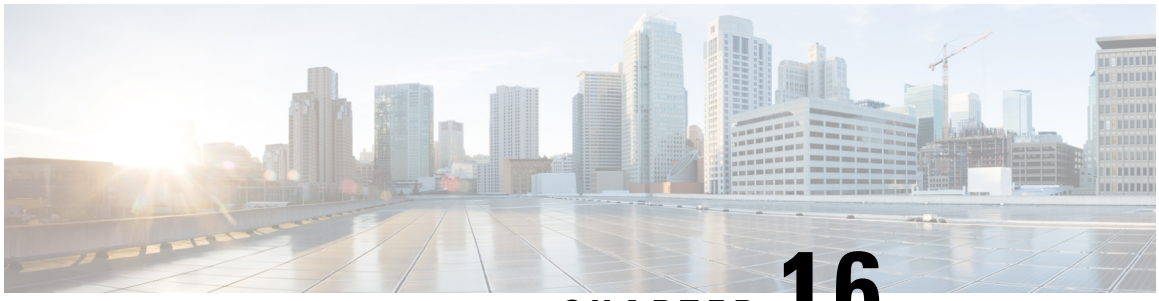
Feature Information for PPPoE over BDI

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 15: Feature Information for PPPoE over BDI

Feature Name	Releases	Feature Information
PPPoE over BDI	Cisco IOS XE Denali 16.3.1.	The PPPoE over BDI feature terminates PPPoE subscribers through a VXLAN L2 overlay network onto a Cisco Bride Domain Interface (BDI). The following commands were modified: pppoe enable group.



CHAPTER 16

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 www.cisco.com/go/cfn. 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

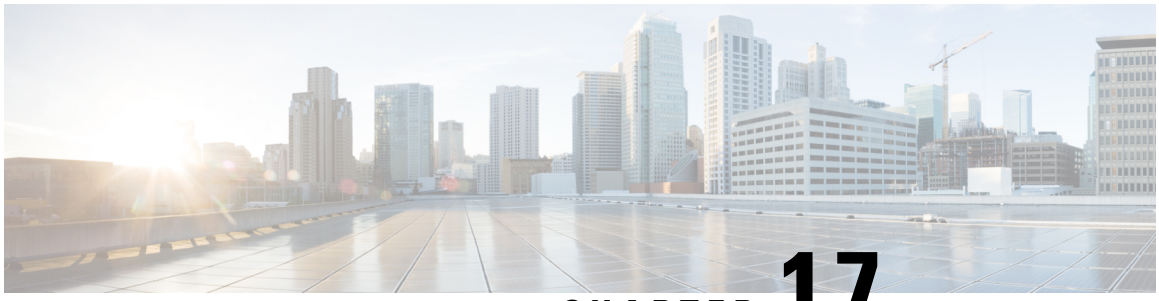
Feature Information for SGT Based PBR

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16: Feature Information for SGT Based PBR

Feature Name	Releases	Feature Information
SGT Based PBR		**This feature is supported on Cisco 4000 Series ISRs.**
SGT Based PBR		The SGT Based PBR feature supports classification of packets based on Security Group Tag (SGT) for grouping the traffic into roles to match the defined policies in PBR. The following commands were introduced or modified: **interface**, **ip policy route-map**, **match security-group destination tag**, **match security-group source tag**, **route-map**, **show ip policy**, **show route-map**, **show route-map dynamic**, **show platform hardware qfp active classification class-group-manager class-group client pbr**, **show platform hardware qfp active classification feature-manager class-group tcam pbr global details**, **match security-group source tag**, **show platform hardware qfp active feature pbr class-group**, **show platform software pbr fp interface all, show platform software pbr rp ac statistics**, **show platform software route-map fp active map**, **show platform software route-map rp active map**.



CHAPTER 17

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

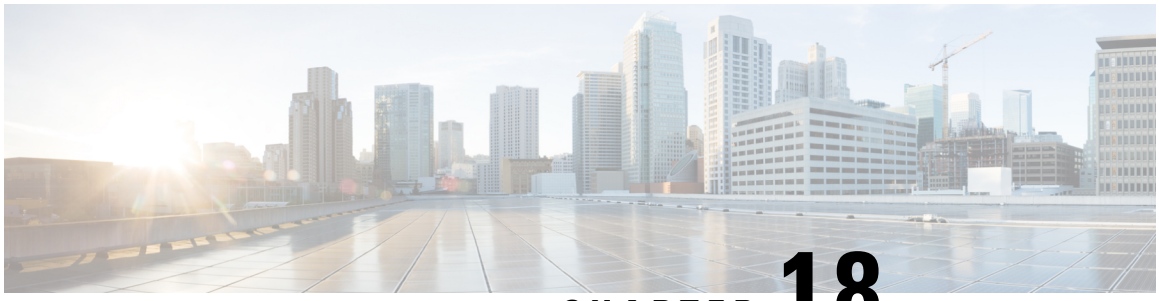
Feature Information for SGT Based QoS

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 17: Feature Information for SGT Based QoS

Feature Name	Releases	Feature Information
SGT Based QoS		The SGT Based QoS feature supports classification of packets based on Security Group Tag (SGT) for grouping the traffic into user groups and devices to match the defined QoS policies. The following commands were introduced or modified: **debug cpl provisioning**, **class-map match security-group destination tag**, **match security-group source tag**, **show class-map**.



CHAPTER 18

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 18: 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

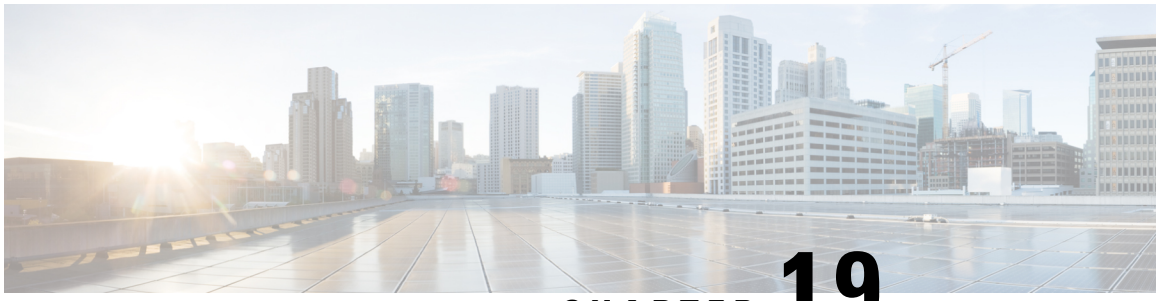
Feature Information for Policy-Based Routing Default Next-Hop Routes

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 19: Feature Information for Policy-Based Routing Default Next-Hop Routes

Feature Name	Releases	Feature Information
Policy-Based Routing Default Next-Hop Routes	12.1(11)E Cisco IOS XE Release 2.2	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 releases, the packets to be forwarded that were generated from the route map for policy-based routing were switched at the software level. The following command was introduced or modified: **set ip default next-hop**.



CHAPTER 19

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

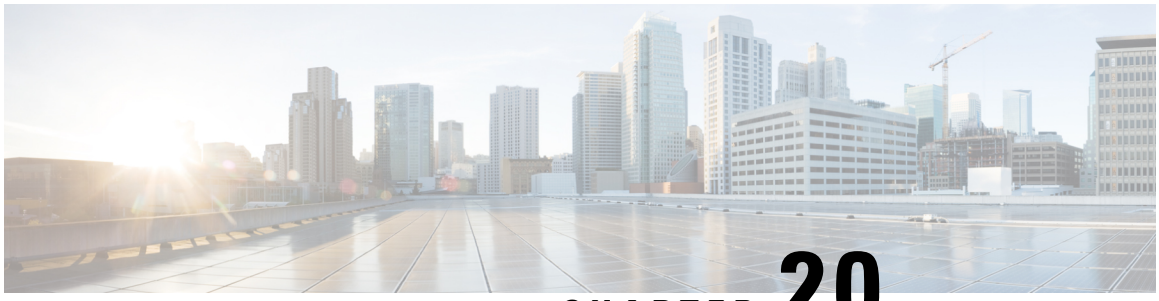
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, 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.

Feature Name	Releases	Feature Information
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.



CHAPTER 20

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.

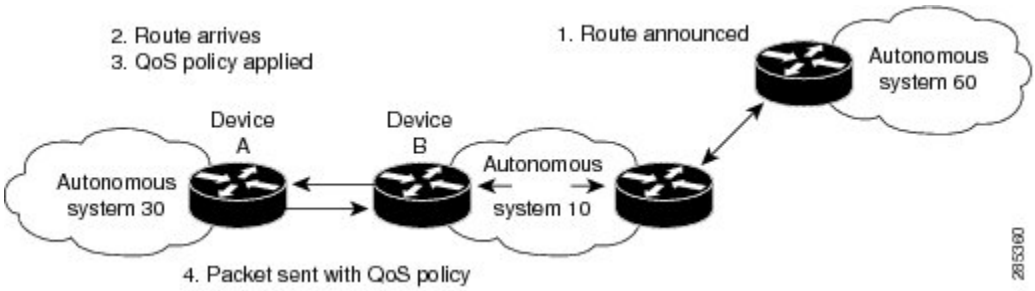
Configuration Examples for QoS Policy Propagation via BGP

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 9: 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
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
IP routing protocol-independent commands	Cisco IOS IP Routing: Protocol-Independent Command Reference
BGP configuration	*BGP Configuration Guide*
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

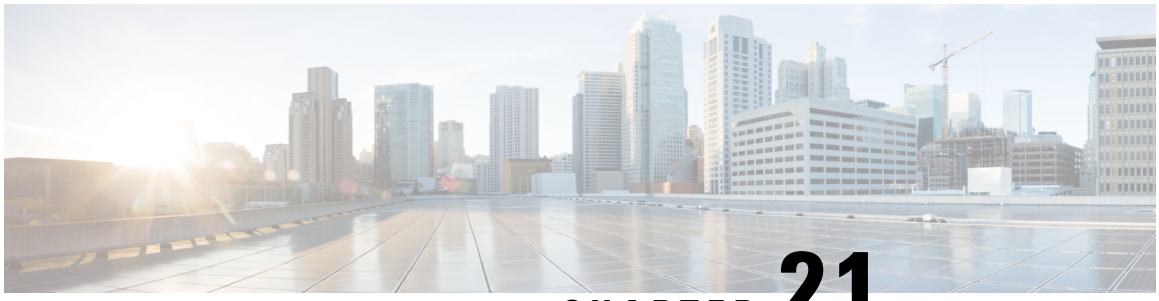
Feature Information for QoS Policy Propagation via BGP

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 20: Feature Information for QoS Policy Propagation via BGP

Feature Name	Releases	Feature Information
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.
Policy Routing Infrastructure		The Policy Routing Infrastructure feature provides full support of IP policy-based routing with Cisco Express Forwarding (CEF). As CEF gradually obsoletes fast switching, policy routing is integrated with CEF to increase customer performance requirements. When policy routing is enabled, redundant processing is avoided.



CHAPTER 21

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

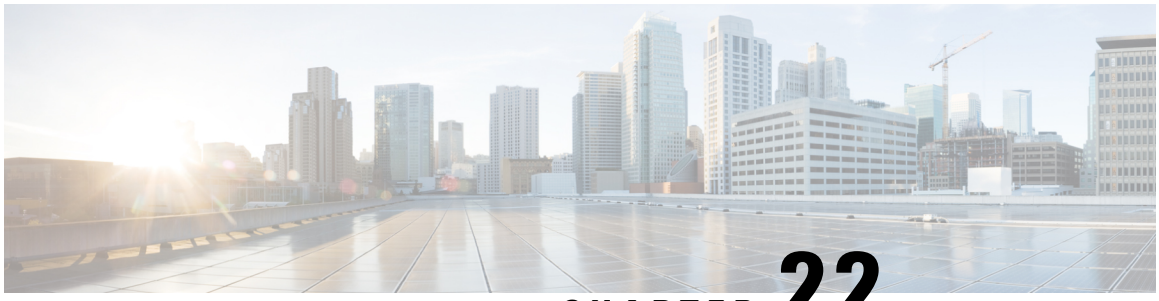
Feature Information for NetFlow Policy Routing

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 21: Feature Information for NetFlow Policy Routing

Feature Name	Releases	Feature Information
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 monitoring information on real-time traffic flows. IP policy routing works with Cisco Express Forwarding, distributed Cisco Express Forwarding, and NetFlow.
Policy Routing Infrastructure		The Policy Routing Infrastructure feature provides full support of IP policy-based routing with Cisco Express Forwarding and NetFlow. When both policy routing and NetFlow are enabled, redundant processing is avoided.



CHAPTER 22

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

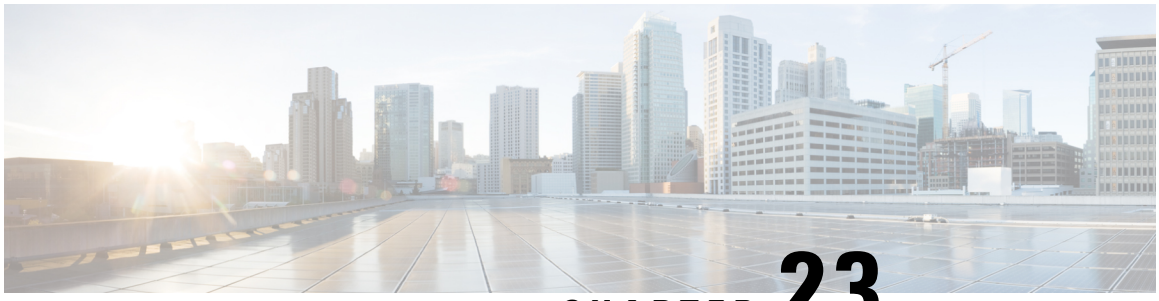
Feature Information for Recursive Static Routes

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 22: Feature Information for Recursive Static Routes

Feature Name	Releases	Feature Information
Recursive Static Routes	Cisco IOS XE Release 3.9S	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. The following command was introduced: **ip route static install-recurse-via-nexthop**.



CHAPTER 23

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 master 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 master 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.

Master Key Tuples

The key chain and keys are used to create Master Key Tuples (MKTs) that are optimized for look-ups during TCP send and receive operations. An MKT consists of a master 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 MKTs.

- current-key: Identifies the MKT that is being used to compute traffic keys for outgoing TCP segments.
- next-key: Identifies the MKT 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 MKT 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 MKT 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 MKT that was used to generate the traffic keys.
- RNextKeyID: The receive identifier of the MKT 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 MKT to be used. On receiving the TCP segment, the peer router, say Router B, finds the MKT 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 MKT. After Router B has rolled over, Router A also sets the current key to the new MKT.

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 master-key for deriving traffic keys.

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 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 23: Feature Information for TCP Authentication Option

Feature Name	Releases	Feature Information
TCP Authentication Option	Cisco IOS XE Gibraltar 16.12.1	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). The following commands were introduced or modified: **key chain** *key-chain-name* **tcp**, **show key chain**, and **show tcp tcb**.