IP Routing: RIP Configuration Guide, Cisco IOS XE Gibraltar 16.11.x

CONTENTS

CHAPTER 4   **Configuring Routing Information Protocol**   **19**

Read Me First

Important Information

Note For CUBE feature support information in Cisco IOS XE Bengaluru 17.6.1a and later releases, see Cisco Unified Border Element IOS-XE Configuration Guide.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

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.

Short Description

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

IPv6 Routing: RIP for IPv6

IPv6 Routing Information Protocol (RIP) functions the same and offers the same benefits as IPv4 RIP. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes and the use of the all-RIP-devices multicast group address, FF02::9, as the destination address for RIP update messages.

Finding Feature Information

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

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

Information About RIP for IPv6

RIP for IPv6

IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages.

In the Cisco software implementation of IPv6 RIP, each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB). The IPv6 RIP RIB contains a set of best-cost IPv6 RIP routes learned from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it will store only the lowest cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors running RIP. IPv6 RIP will try to insert every non-expired route from its local RIB into the primary IPv6 RIB. If the same route has been learned

from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route will not be added to the IPv6 RIB but the RIP route will still exist in the IPv6 RIP RIB.

Nonstop Forwarding for IPv6 RIP

Cisco nonstop forwarding (NSF) continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. When an RP failover occurs, the Forwarding Information Base (FIB) marks installed paths as stale by setting a new epoch. Subsequently, the routing protocols reconverge and populate the RIB and FIB. Once all NSF routing protocols converge, any stale routes held in the FIB are removed. A failsafe timer is required to delete stale routes, in case of routing protocol failure to repopulate the RIB and FIB.

RIP registers as an IPv6 NSF client. Doing so has the benefit of using RIP routes installed in the Cisco Express Forwarding table until RIP has converged on the standby.

How to Configure RIP for IPv6

Enabling IPv6 RIP

Before you begin

Before configuring the router to run IPv6 RIP, globally enable IPv6 using the **ipv6 unicast-routing** command in global configuration mode, and enable IPv6 on any interfaces on which IPv6 RIP is to be enabled.

If you want to set or change a global value, follow steps 1 and 2, and then use the optional **ipv6 router rip**command in global configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 rip** *name* **enable**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 5	**ipv6 rip** *name* **enable** **Example:** `Router(config-if)# ipv6 rip process1 enable`	Enables the specified IPv6 RIP routing process on an interface.

Customizing IPv6 RIP

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router rip** *word*
4. **maximum-paths** *number-paths*
5. **exit**
6. **interface** *type number*
7. **ipv6 rip** *name* **default-information** {**only** | **originate**} [**metric** *metric-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router rip** *word* **Example:** `Router(config)# ipv6 router rip process1`	Configures an IPv6 RIP routing process and enters router configuration mode for the IPv6 RIP routing process. • Use the *word* argument to identify a specific IPv6 RIP routing process.

	Command or Action	**Purpose**
Step 4	**maximum-paths** *number-paths* **Example:** `Router(config-router)# maximum-paths 1`	(Optional) Defines the maximum number of equal-cost routes that IPv6 RIP can support. • The *number-paths* argument is an integer from 1 to 64. The default for RIP is four paths.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 7	**ipv6 rip** *name* **default-information** {**only** | **originate**} [**metric** *metric-value*] **Example:** `Router(config-if)# ipv6 rip process1 default-information originate`	(Optional) Originates the IPv6 default route (::/0) into the specified RIP routing process updates sent out of the specified interface. **Note** To avoid routing loops after the IPv6 default route (::/0) is originated out of any interface, the routing process ignores all default routes received on any interface. • Specifying the **only** keyword originates the default route (::/0) but suppresses all other routes in the updates sent on this interface. • Specifying the **originate** keyword originates the default route (::/0) in addition to all other routes in the updates sent on this interface.

Verifying IPv6 RIP Configuration and Operation

SUMMARY STEPS

1. **show ipv6 rip** [*name*][**database**\| **next-hops**]
2. **show ipv6 route** [*ipv6-address*\| *ipv6-prefix/prefix-length*\| *protocol* \| *interface-type interface-number*]
3. **enable**
4. **debug ipv6 rip** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**show ipv6 rip** [*name*][**database**| **next-hops**] **Example:**	(Optional) Displays information about current IPv6 RIP processes.

	Command or Action	Purpose
	`Device> show ipv6 rip process1 database`	• In this example, IPv6 RIP process database information is displayed for the specified IPv6 RIP process.
Step 2	**show ipv6 route** [*ipv6-address*| *ipv6-prefix*/*prefix-length*| *protocol* | *interface-type interface-number*] **Example:** `Device> show ipv6 route rip`	(Optional) Displays the current contents of the IPv6 routing table. • In this example, only IPv6 RIP routes are displayed.
Step 3	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 4	**debug ipv6 rip** [*interface-type interface-number*] **Example:** `Device# debug ipv6 rip`	(Optional) Displays debugging messages for IPv6 RIP routing transactions.

Configuration Examples for RIP for IPv6

Example: Enabling the RIP for IPv6 Process

In the following example, the IPv6 RIP process named process1 is enabled on the router and on Gigabit Ethernet interface 0/0/0. The IPv6 default route (::/0) is advertised in addition to all other routes in router updates sent on Gigabit Ethernet interface 0/0/0. Additionally, BGP routes are redistributed into the RIP process named process1 according to a route map where routes that match a prefix list are also tagged. The number of parallel paths is set to one to allow the route tagging, and the IPv6 RIP timers are adjusted. A prefix list named eth0/0-in-flt filters inbound routing updates on Gigabit Ethernet interface 0/0/0.

```
ipv6 router rip process1
 maximum-paths 1
 redistribute bgp 65001 route-map bgp-to-rip
 distribute-list prefix-list eth0/0-in-flt in Gigabitethernet0/0/0
!
interface Gigabitethernet0/0/0
 ipv6 address 2001:DB8::/64 eui-64
 ipv6 rip process1 enable
 ipv6 rip process1 default-information originate
!
ipv6 prefix-list bgp-to-rip-flt seq 10 deny 2001:DB8:3::/16 le 128
ipv6 prefix-list bgp-to-rip-flt seq 20 permit 2001:DB8:1::/8 le 128
!
ipv6 prefix-list eth0/0-in-flt seq 10 deny ::/0
ipv6 prefix-list eth0/0-in-flt seq 15 permit ::/0 le 128
!
route-map bgp-to-rip permit 10
```

```
 match ipv6 address prefix-list bgp-to-rip-flt
 set tag 4
```

In the following example, output information about all current IPv6 RIP processes is displayed using the **show ipv6 rip** command:

```
Device> show ipv6 rip

RIP process "process1", port 521, multicast-group FF02::9, pid 62
    Administrative distance is 120. Maximum paths is 1
    Updates every 5 seconds, expire after 15
    Holddown lasts 10 seconds, garbage collect after 30
    Split horizon is on; poison reverse is off
    Default routes are generated
    Periodic updates 223, trigger updates 1
  Interfaces:
    Gigabitethernet0/0/0
  Redistribution:
    Redistributing protocol bgp 65001 route-map bgp-to-rip
```

In the following example, output information about a specified IPv6 RIP process database is displayed using the **show ipv6 rip** command with the *name* argument and the **database** keyword. In the following output for the IPv6 RIP process named process1, timer information is displayed, and route 2001:DB8::16/64 has a route tag set:

```
Device> show ipv6 rip process1 database

RIP process "process1", local RIB
 2001:DB8::/64, metric 2
    Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8::/16, metric 2 tag 4, installed
    Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:1::/16, metric 2 tag 4, installed
    Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:2::/16, metric 2 tag 4, installed
    Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 ::/0, metric 2, installed
    Gigabitethernet0/0/0FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
```

In the following example, output information for a specified IPv6 RIP process is displayed using the **show ipv6 rip** command with the *name* argument and the **next-hops** keyword:

```
Device> show ipv6 rip process1 next-hops

RIP process "process1", Next Hops
  FE80::A8BB:CCFF:FE00:A00/Gigabitethernet0/0/0 [4 paths]
```

Additional References

The following sections provide references related to configuring Routing Information Protocol.

Related Documents

Related Topic	Document Title
Protocol-independent features, filtering RIP information, key management (available in RIP Version 2), and VLSM	*Configuring IP Routing Protocol-Independent Features*
IPv6 Routing: RIP for IPv6	*Cisco IOS IP Routing: RIP Configuration Guide*

Related Topic	Document Title
RIP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: RIP Command Reference*
Configuring Frame Relay	*Cisco IOS Wide-Area Networking Configuration Guide*

Standards

Standard	Title
None	--

MIBs

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

RFCs

RFC	Title
RFC 1058	*Routing Information Protocol*
RFC 2082	RIP-2 MD5 Authentication
RFC 2091	*Triggered Extensions to RIP to Support Demand Circuits*
RFC 2453	RIP version 2

Technical Assistance

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

Feature Information for RIP for IPv6

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

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

Table 1: Feature Information for RIP for IPv6

Feature Name	Releases	Feature Information
IPv6 Routing: RIP for IPv6 (RIPng)	12.2(2)T 12.2(17a)SX1 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 12.3 15.0(2)SG Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2.0SG	RIP enhancements for IPv6 include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages. The following commands were introduced or modified: **debug ipv6 rip**, **ipv6 rip default-information**, **ipv6 rip enable**, **ipv6 router rip**, **ipv6 unicast-routing**, **maximum-paths**, **show ipv6 rip**, **show ipv6 route**.
IPv6: RIPng Nonstop Forwarding	12.2(33)SRE 15.0(1)S 15.0(1)SY Cisco IOS XE Release 2.1	The IPv6 RIPng nonstop forwarding feature is supported.

IPv6 Routing: Route Redistribution

IPv6 route redistribution allows routes to be specified by prefix, using a route-map prefix list, or by tag, using the route-map "match tag" function.

Finding Feature Information

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

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

Information About IPv6 Route Redistribution

RIP for IPv6

IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages.

In the Cisco software implementation of IPv6 RIP, each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB). The IPv6 RIP RIB contains a set of best-cost IPv6 RIP routes learned from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it will store only the lowest cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors running RIP. IPv6 RIP will try to insert every non-expired route from its local RIB into the primary IPv6 RIB. If the same route has been learned

from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route will not be added to the IPv6 RIB but the RIP route will still exist in the IPv6 RIP RIB.

How to Configure IPv6 Route Redistribution

Redistributing Routes into an IPv6 RIP Routing Process

The maximum metric that RIP can advertise is 16, and a metric of 16 denotes a route that is unreachable. Therefore, if you are redistributing routes with metrics greater than or equal to 16, then by default RIP will advertise them as unreachable. These routes will not be used by neighboring routers. The user must configure a redistribution metric of less than 15 for these routes.

> ✎
>
> **Note** You must to advertise a route with metric of 15 or less. A RIP router always adds an interface cost--the default is 1--onto the metric of a received route. If you advertise a route with metric 15, your neighbor will add 1 to it, making a metric of 16. Because a metric of 16 is unreachable, your neighbor will not install the route in the routing table.

If no metric is specified, then the current metric of the route is used. To find the current metric of the route, enter the **show ipv6 route** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 rip** *word* **enable**
5. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2**| **level-2**} [**metric** *metric-value*] [**metric-type**{**internal** | **external**}] [**route-map** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
	`Router(config)# interface gigabitethernet 0/0/0`	
Step 4	**ipv6 rip** *word* **enable** **Example:** `Router(config-if)# ipv6 router one enable`	Enables an IPv6 Routing Information Protocol (RIP) routing process on an interface.

Step 5	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2**| **level-2**} [**metric** *metric-value*] [**metric-type**{**internal** | **external**}] [**route-map** *map-name*]
	
	Example:
	
	`Router(config-router)# redistribute bgp 65001`
	`route-map bgp-to-rip`
		Redistributes the specified routes into the IPv6 RIP routing process.
		
		• The *protocol* argument can be one of the following keywords: **bgp**, **connected**, **isis**, **rip**, or **static**.
		
		• The **rip** keyword and *process-id* argument specify an IPv6 RIP routing process.
		
		Note The **connected** keyword refers to routes that are established automatically by assigning IPv6 addresses to an interface.

Configuring Route Tags for IPv6 RIP Routes

When performing route redistribution, you can associate a numeric tag with a route. The tag is advertised with the route by RIP and will be installed along with the route in neighboring router's routing table.

If you redistribute a tagged route (for example, a route in the IPv6 routing table that already has a tag) into RIP, then RIP will automatically advertise the tag with the route. If you use a redistribution route map to specify a tag, then RIP will use the route map tag in preference to the routing table tag.

SUMMARY STEPS

1. **enable**
2. **configure   terminal**
3. **route-map**   *map-tag* [**permit** \| **deny**] [*sequence-number*]
4. **match ipv6 address**   {**prefix-list** *prefix-list-name* \| *access-list-name*
5. **set tag**   *tag-value*

DETAILED STEPS

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

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map bgp-to-rip permit 10`	Defines a route map, and enters route-map configuration mode. • Follow this step with a **match** command.
Step 4	**match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name* **Example:** `Router(config-route-map)# match ipv6 address prefix-list bgp-to-rip-flt`	Specifies a list of IPv6 prefixes to be matched.
Step 5	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 4`	Sets the tag value to associate with the redistributed routes.

Filtering IPv6 RIP Routing Updates

Route filtering using distribute lists provides control over the routes RIP receives and advertises. This control may be exercised globally or per interface.

Filtering is controlled by distribute lists. Input distribute lists control route reception, and input filtering is applied to advertisements received from neighbors. Only those routes that pass input filtering will be inserted in the RIP local routing table and become candidates for insertion into the IPv6 routing table.

Output distribute lists control route advertisement; Output filtering is applied to route advertisements sent to neighbors. Only those routes passing output filtering will be advertised.

Global distribute lists (which are distribute lists that do not apply to a specified interface) apply to all interfaces. If a distribute list specifies an interface, then that distribute list applies only to that interface.

An interface distribute list always takes precedence. For example, for a route received at an interface, with the interface filter set to deny, and the global filter set to permit, the route is blocked, the interface filter is passed, the global filter is blocked, and the route is passed.

IPv6 prefix lists are used to specify certain prefixes or a range of prefixes that must be matched before a permit or deny statement can be applied. Two operand keywords can be used to designate a range of prefix lengths to be matched. A prefix length of less than, or equal to, a value is configured with the **le**keyword. A prefix length greater than, or equal to, a value is specified using the **ge** keyword. The **ge** and **le** keywords can be used to specify the range of the prefix length to be matched in more detail than the usual *ipv6-prefix* / *prefix-length* argument. For a candidate prefix to match against a prefix list entry three conditions can exist:

• The candidate prefix must match the specified prefix list and prefix length entry.

• The value of the optional **le** keyword specifies the range of allowed prefix lengths from the *prefix-length* argument up to, and including, the value of the **le** keyword.

- The value of the optional **ge** keyword specifies the range of allowed prefix lengths from the value of the **ge** keyword up to, and including, 128.

✎

Note   Note that the first condition must match before the other conditions take effect.

An exact match is assumed when the **ge** or **le** keywords are not specified. If only one keyword operand is specified then the condition for that keyword is applied, and the other condition is not applied. The *prefix-length* value must be less than the **ge** value. The **ge** value must be less than, or equal to, the **le** value. The **le** value must be less than or equal to 128.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*
4. **ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*
5. Repeat Steps 3 and 4 as many times as necessary to build the prefix list.
6. **ipv6 router rip** *name*
7. **distribute-list prefix-list** *prefix-list-name* **in** | **out**} [*interface-type interface-number*

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length*	**description** *text*} [**ge** *ge-value*] [**le** *le-value* **Example:** `Router(config)# ipv6 prefix-list abc permit 2001:DB8::/16`	Creates an entry in the IPv6 prefix list.
Step 4	**ipv6 prefix list** *prefix-list-name* **seq** *seq-number*] {**deny** *ipv6-prefix*/*prefix-length*	**description** *text*} [**ge** *ge-value*] [**le** *le-value* **Example:**	Creates an entry in the IPv6 prefix list.

	Command or Action	Purpose
	`Router(config)# ipv6 prefix-list abc deny ::/0`	
Step 5	Repeat Steps 3 and 4 as many times as necessary to build the prefix list.	--
Step 6	**ipv6 router rip** *name* **Example:** `Router(config)# ipv6 router rip process1`	Configures an IPv6 RIP routing process.

Step 7	**distribute-list prefix-list** *prefix-list-name* **in** | **out**}
	[*interface-type interface-number*
	
	Example:
	
	`Router(config-rtr-rip)# distribute-list prefix-list`
	`process1 in gigabitethernet 0/0/0`
		Applies a prefix list to IPv6 RIP routing updates that are received or sent on an interface.

Configuration Examples for IPv6 Route Redistribution

Example: Enabling the RIP for IPv6 Process

In the following example, the IPv6 RIP process named process1 is enabled on the router and on Gigabit Ethernet interface 0/0/0. The IPv6 default route (::/0) is advertised in addition to all other routes in router updates sent on Gigabit Ethernet interface 0/0/0. Additionally, BGP routes are redistributed into the RIP process named process1 according to a route map where routes that match a prefix list are also tagged. The number of parallel paths is set to one to allow the route tagging, and the IPv6 RIP timers are adjusted. A prefix list named eth0/0-in-flt filters inbound routing updates on Gigabit Ethernet interface 0/0/0.

```
ipv6 router rip process1
 maximum-paths 1
 redistribute bgp 65001 route-map bgp-to-rip
 distribute-list prefix-list eth0/0-in-flt in Gigabitethernet0/0/0
!
interface Gigabitethernet0/0/0
 ipv6 address 2001:DB8::/64 eui-64
 ipv6 rip process1 enable
 ipv6 rip process1 default-information originate
!
ipv6 prefix-list bgp-to-rip-flt seq 10 deny 2001:DB8:3::/16 le 128
ipv6 prefix-list bgp-to-rip-flt seq 20 permit 2001:DB8:1::/8 le 128
!
ipv6 prefix-list eth0/0-in-flt seq 10 deny ::/0
ipv6 prefix-list eth0/0-in-flt seq 15 permit ::/0 le 128
!
route-map bgp-to-rip permit 10
 match ipv6 address prefix-list bgp-to-rip-flt
 set tag 4
```

In the following example, output information about all current IPv6 RIP processes is displayed using the **show ipv6 rip** command:

```
Device> show ipv6 rip

RIP process "process1", port 521, multicast-group FF02::9, pid 62
     Administrative distance is 120. Maximum paths is 1
     Updates every 5 seconds, expire after 15
     Holddown lasts 10 seconds, garbage collect after 30
     Split horizon is on; poison reverse is off
     Default routes are generated
     Periodic updates 223, trigger updates 1
  Interfaces:
    Gigabitethernet0/0/0
  Redistribution:
    Redistributing protocol bgp 65001 route-map bgp-to-rip
```

In the following example, output information about a specified IPv6 RIP process database is displayed using the **show ipv6 rip** command with the *name* argument and the **database** keyword. In the following output for the IPv6 RIP process named process1, timer information is displayed, and route 2001:DB8::16/64 has a route tag set:

```
Device> show ipv6 rip process1 database

RIP process "process1", local RIB
 2001:DB8::/64, metric 2
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:1::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:DB8:2::/16, metric 2 tag 4, installed
     Gigabitethernet0/0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 ::/0, metric 2, installed
     Gigabitethernet0/0/0FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
```

In the following example, output information for a specified IPv6 RIP process is displayed using the **show ipv6 rip** command with the *name* argument and the **next-hops** keyword:

```
Device> show ipv6 rip process1 next-hops

RIP process "process1", Next Hops
  FE80::A8BB:CCFF:FE00:A00/Gigabitethernet0/0/0 [4 paths]
```

Additional References

Related Documents

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

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

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

Technical Assistance

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

Feature Information for IPv6 Routing: Route Redistribution

Table 2: Feature Information for IPv6 Routing: Route Redistribution

Feature Name	Releases	Feature Information
IPv6 Routing: Route Redistribution	Cisco IOS XE Release 2.1	Routes may be specified by prefix, using a route-map prefix list, or by tag, using the route-map "match tag" function. The following commands were introduced or modified: **distribute-list prefix-list**, **ipv6 prefix list**, **ipv6 rip enable**, **ipv6 router rip**, **match ipv6 address**, **redistribute**, **route-map**, **set tag**, **show ipv6 rip**.

Configuring Routing Information Protocol

Routing Information Protocol (RIP) is a commonly used routing protocol in small to medium TCP/IP networks. It is a stable protocol that uses a distance-vector algorithm to calculate routes.

Finding Feature Information

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

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

Prerequisites for RIP

You must configure **ip routing** command before you configure RIP.

Restrictions for RIP

Routing Information Protocol (RIP) uses hop count as the metric to rate the value of different routes. The hop count is the number of devices that can be traversed in a route. A directly connected network has a metric of zero; an unreachable network has a metric of 16. This limited metric range makes RIP unsuitable for large networks.

Note   If RIP configuration does not have a network statement covering a specific interface, we recommend that you do not configure RIP under that interface. If RIP is configured on such an interface, the redistribution of route(s) from another routing protocol into RIP, received through that interface, does not work.

Information About Configuring RIP

RIP Overview

The Routing Information Protocol (RIP) uses broadcast UDP data packets to exchange routing information. Cisco software sends routing information updates every 30 seconds, which is termed advertising. If a device does not receive an update from another device for 180 seconds or more, the receiving device marks the routes served by the nonupdating device as unusable. If there is still no update after 240 seconds, the device removes all routing table entries for the nonupdating device.

A device that is running RIP can receive a default network via an update from another device that is running RIP, or the device can source the default network using RIP. In both cases, the default network is advertised through RIP to other RIP neighbors.

The Cisco implementation of RIP Version 2 (RIPv2) supports plain text and message digest algorithm 5 (MD5) authentication, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs).

RIP Routing Updates

The Routing Information Protocol (RIP) sends routing-update messages at regular intervals and when the network topology changes. When a device receives a RIP routing update that includes changes to an entry, the device updates its routing table to reflect the new route. The metric value for the path is increased by 1, and the sender is indicated as the next hop. RIP devices maintain only the best route (the route with the lowest metric value) to a destination. After updating its routing table, the device immediately begins transmitting RIP routing updates to inform other network devices of the change. These updates are sent independently of the regularly scheduled updates that RIP devices send.

RIP Routing Metric

The Routing Information Protocol (RIP) uses a single routing metric to measure the distance between the source and the destination network. Each hop in a path from the source to the destination is assigned a hop-count value, which is typically 1. When a device receives a routing update that contains a new or changed destination network entry, the device adds 1 to the metric value indicated in the update and enters the network in the routing table. The IP address of the sender is used as the next hop. If an interface network is not specified in the routing table, it will not be advertised in any RIP update.

Authentication in RIP

The Cisco implementation of the Routing Information Protocol (RIP) Version 2 (RIPv2) supports authentication, key management, route summarization, classless interdomain routing (CIDR), and variable-length subnet masks (VLSMs).

By default, the software receives RIP Version 1 (RIPv1) and RIPv2 packets, but sends only RIPv1 packets. You can configure the software to receive and send only RIPv1 packets. Alternatively, you can configure the software to receive and send only RIPv2 packets. To override the default behavior, you can configure the RIP version that an interface sends. Similarly, you can also control how packets received from an interface are processed.

RIPv1 does not support authentication. If you are sending and receiving RIP v2 packets, you can enable RIP authentication on an interface.

The key chain determines the set of keys that can be used on the interface. Authentication, including default authentication, is performed on that interface only if a key chain is configured. For more information on key chains and their configuration, see the "Managing Authentication Keys" section in the "Configuring IP Routing Protocol-Independent Features" chapter in the *Cisco IOS IP Routing: Protocol-Independent Configuration Guide*.

Cisco supports two modes of authentication on an interface on which RIP is enabled: plain-text authentication and message digest algorithm 5 (MD5) authentication. Plain-text authentication is the default authentication in every RIPv2 packet.

Note Do not use plain text authentication in RIP packets for security purposes, because the unencrypted authentication key is sent in every RIPv2 packet. Use plain-text authentication when security is not an issue; for example, you can use plain-text authentication to ensure that misconfigured hosts do not participate in routing.

Exchange of Routing Information

Routing Information Protocol (RIP) is normally a broadcast protocol, and for RIP routing updates to reach nonbroadcast networks, you must configure the Cisco software to permit this exchange of routing information.

To control the set of interfaces with which you want to exchange routing updates, you can disable the sending of routing updates on specified interfaces by configuring the **passive-interface** router configuration command.

You can use an offset list to increase increasing incoming and outgoing metrics to routes learned via RIP. Optionally, you can limit the offset list with either an access list or an interface.

Routing protocols use several timers that determine variables such as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs. You can make the following timer adjustments:

- The rate (time, in seconds, between updates) at which routing updates are sent
- The interval of time, in seconds, after which a route is declared invalid
- The interval, in seconds, during which routing information about better paths is suppressed
- The amount of time, in seconds, that must pass before a route is removed from the routing table
- The amount of time for which routing updates will be postponed

You can adjust the IP routing support in the Cisco software to enable faster convergence of various IP routing algorithms, and hence, cause quicker fallback to redundant devices. The total effect is to minimize disruptions to end users of the network in situations where quick recovery is essential

In addition, an address family can have timers that explicitly apply to that address family (or Virtual Routing and Forwarding [VRF]) instance). The **timers-basic** command must be specified for an address family or the system defaults for the **timers-basic** command are used regardless of the timer that is configured for RIP routing. The VRF does not inherit the timer values from the base RIP configuration. The VRF will always use the system default timers unless the timers are explicitly changed using the **timers-basic** command.

RIP Route Summarization

Summarizing routes in RIP Version 2 improves scalability and efficiency in large networks. Summarizing IP addresses means that there is no entry for child routes (routes that are created for any combination of the individual IP addresses contained within a summary address) in the RIP routing table, reducing the size of the table and allowing the router to handle more routes.

Summary IP address functions more efficiently than multiple individually advertised IP routes for the following reasons:

- The summarized routes in the RIP database are processed first.

- Any associated child routes that are included in a summarized route are skipped as RIP looks through the routing database, reducing the processing time required. Cisco routers can summarize routes in two ways:

- Automatically, by summarizing subprefixes to the classful network boundary when crossing classful network boundaries (automatic summary).

Note   Automatic summary is enabled by default.

- As specifically configured, advertising a summarized local IP address pool on the specified interface (on a network access server) so that the address pool can be provided to dialup clients.

When RIP determines that a summary address is required in the RIP database, a summary entry is created in the RIP routing database. As long as there are child routes for a summary address, the address remains in the routing database. When the last child route is removed, the summary entry also is removed from the database. This method of handling database entries reduces the number of entries in the database because each child route is not listed in an entry, and the aggregate entry itself is removed when there are no longer any valid child routes for it.

RIP Version 2 route summarization requires that the lowest metric of the "best route" of an aggregated entry, or the lowest metric of all current child routes, be advertised. The best metric for aggregated summarized routes is calculated at route initialization or when there are metric modifications of specific routes at advertisement time, and not at the time the aggregated routes are advertised.

The **ip summary-address rip router**configuration command causes the router to summarize a given set of routes learned via RIP Version 2 or redistributed into RIP Version 2. Host routes are especially applicable for summarization.

See the "Route Summarization Example, on page 40" section at the end of this chapter for examples of using split horizon.

You can verify which routes are summarized for an interface using the **show ip protocols** EXEC command. You can check summary address entries in the RIP database. These entries will appear in the database only if relevant child routes are being summarized. To display summary address entries in the RIP routing database entries if there are relevant routes being summarized based upon a summary address, use the **show ip rip database** command in EXEC mode. When the last child route for a summary address becomes invalid, the summary address is also removed from the routing table.

Split Horizon Mechanism

Normally, devices that are connected to broadcast-type IP networks and that use distance-vector routing protocols employ the split horizon mechanism to reduce the possibility of routing loops. The split horizon mechanism blocks information about routes from being advertised by a device out of any interface from which that information originated. This behavior usually optimizes communications among multiple devices, particularly when links are broken. However, with nonbroadcast networks, such as Frame Relay and the Switched Multimegabit Digital System (SMDS), situations can arise for which this behavior is less than ideal. In such situations, you may want to disable split horizon with the Routing Information Protocol (RIP).

If an interface is configured with secondary IP addresses and split horizon is enabled, updates might not be sourced by the secondary address. If split horizon is enabled, one routing update is sourced per network number.

Split horizon is not disabled by default for interfaces using any of the X.25 encapsulations. For all other encapsulations, split horizon is enabled by default.

Interpacket Delay for RIP Updates

By default, the software adds no delay between packets in a multiple-packet RIP update being sent. If you have a high-end router sending to a low-speed router, you might want to add such interpacket delay to RIP updates, in the range of 8 to 50 milliseconds.

RIP Optimization over WAN Circuits

Devices are used on connection-oriented networks to allow potential connectivity to many remote destinations. Circuits on the WAN are established on demand and are relinquished when the traffic subsides. Depending on the application, the connection between any two sites for user data could be short and relatively infrequent.

Source IP Addresses of RIP Routing Updates

By default, the Cisco software validates the source IP address of incoming Routing Information Protocol (RIP) routing updates. If the source address is not valid, the software discards the routing update. You must disable this functionality if you want to receive updates from a device that is not part of this network. However, disabling this functionality is not recommended under normal circumstances.

Neighbor Router Authentication

You can prevent your router from receiving fraudulent route updates by configuring neighbor router authentication. When configured, neighbor authentication occurs whenever routing updates are exchanged between neighbor routers. This authentication ensures that a router receives reliable routing information from a trusted source.

Without neighbor authentication, unauthorized or deliberately malicious routing updates could compromise the security of your network traffic. A security compromise could occur if an unfriendly party diverts or analyzes your network traffic. For example, an unauthorized router could send a fictitious routing update to convince your router to send traffic to an incorrect destination. This diverted traffic could be analyzed to learn confidential information about your organization or merely used to disrupt your organization's ability to effectively communicate using the network. Neighbor authentication prevents any such fraudulent route updates from being received by your router.

When neighbor authentication has been configured on a router, the router authenticates the source of each routing update packet that it receives. This is accomplished by the exchange of an authenticating key (sometimes referred to as a password) that is known to both the sending and the receiving router.

There are two types of neighbor authentication used: plain text authentication and Message Digest Algorithm Version 5 (MD5) authentication. Both forms work in the same way, with the exception that MD5 sends a "message digest" instead of the authenticating key itself. The message digest is created using the key and a message, but the key itself is not sent, preventing it from being read while it is being transmitted. Plain text authentication sends the authenticating key itself over the wire.

> **Note** Note that plain text authentication is not recommended for use as part of your security strategy. Its primary use is to avoid accidental changes to the routing infrastructure. Using MD5 authentication, however, is a recommended security practice.

In plain text authentication, each participating neighbor router must share an authenticating key. This key is specified at each router during configuration. Multiple keys can be specified with some protocols; each key must then be identified by a key number.

In general, when a routing update is sent, the following authentication sequence occurs:

1. A router sends a routing update with a key and the corresponding key number to the neighbor router. In protocols that can have only one key, the key number is always zero. The receiving (neighbor) router checks the received key against the same key stored in its own memory.

2. If the two keys match, the receiving router accepts the routing update packet. If the two keys do not match, the routing update packet is rejected.

MD5 authentication works similarly to plain text authentication, except that the key is never sent over the wire. Instead, the router uses the MD5 algorithm to produce a "message digest" of the key (also called a "hash"). The message digest is then sent instead of the key itself. This ensures that nobody can eavesdrop on the line and learn keys during transmission.

Another form of neighbor router authentication is to configure key management using key chains. When you configure a key chain, you specify a series of keys with lifetimes, and the Cisco IOS software rotates through each of these keys. This decreases the likelihood that keys will be compromised. To find complete configuration information for key chains, refer to the "Managing Authentication Keys" section in the Configuring IP Routing Protocol-Independent Features module of the Cisco IOS IP Routing: Protocol-Independent Configuration Guide.

IP-RIP Delay Start Overview

The IP-RIP Delay Start feature is used on Cisco devices to delay the initiation of Routing Information Protocol Version 2 (RIPv2) neighbor sessions until the network connectivity between the neighbor devices is fully operational, thereby ensuring that the sequence number of the first message digest algorithm 5 (MD5) packet

that the device sends to the non-Cisco neighbor device is 0. The default behavior for a device configured to establish RIPv2 neighbor sessions with a neighbor device using MD5 authentication is to start sending MD5 packets when the physical interface is up.

The IP-RIP Delay Start feature is often used when a Cisco device is configured to establish a RIPv2 neighbor relationship using MD5 authentication with a non-Cisco device over a Frame Relay network. When RIPv2 neighbors are connected over Frame Relay, it is possible for the serial interface connected to the Frame Relay network to be up while the underlying Frame Relay circuits are not yet ready to transmit and receive data. When a serial interface is up and the Frame Relay circuits are not yet operational, any MD5 packets that the device attempts to transmit over the serial interface are dropped. When MD5 packets are dropped because the Frame Relay circuits over which the packets need to be transmitted are not yet operational, the sequence number of the first MD5 packet received by the neighbor device after the Frame Relay circuits become active will be greater than 0. Some non-Cisco devices will not allow an MD5-authenticated RIPv2 neighbor session to start when the sequence number of the first MD5 packet received from the other device is greater than 0.

The differences in vendor implementations of MD5 authentication for RIPv2 are probably a result of the ambiguity of the relevant RFC (RFC 2082) with respect to packet loss. RFC 2082 suggests that devices should be ready to accept either a sequence number of 0 or a sequence number higher than the last sequence number received. For more information about MD5 message reception for RIPv2, see section 3.2.2 of RFC 2082 at the following url: http://www.ietf.org/rfc/rfc2082.txt.

The IP-RIP Delay Start feature is supported over other interface types such as Fast Ethernet and Gigabit Ethernet.

Cisco devices allow an MD5-authenticated RIPv2 neighbor session to start when the sequence number of the first MD5 packet received from the other device is greater than 0. If you are using only Cisco devices in your network, you do not need to use the IP-RIP Delay Start feature.

Offset-list

An offset list is the mechanism for increasing incoming and outgoing metrics to routes learned via RIP. This is done to provide a local mechanism for increasing the value of routing metrics. Optionally, you can limit the offset list with either an access list or an interface.

Timers

Routing protocols use several timers that determine such variables as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs. You can make the following timer adjustments:

   • The rate (time in seconds between updates) at which routing updates are sent

   • The interval of time (in seconds) after which a route is declared invalid

   • The interval (in seconds) during which routing information regarding better paths is suppressed

   • The amount of time (in seconds) that must pass before a route is removed from the routing table

   • The amount of time for which routing updates will be postponed

It also is possible to tune the IP routing support in the software to enable faster convergence of the various IP routing algorithms, and, hence, quicker fallback to redundant routers. The total effect is to minimize disruptions to end users of the network in situations where quick recovery is essential.

How to Configure RIP

Enabling RIP and Configuring RIP Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **network** *ip-address*
5. **neighbor** *ip-address*
6. **offset-list** [*access-list-number* | access-list-*name*] {**in** | **out**} *offset* [*interface-type interface-number*]
7. **timers basic** *update invalid holddown flush* [*sleeptime*]
8. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router rip** **Example:** `Device(config)# router rip`	Enables a RIP routing process and enters router configuration mode.
Step 4	**network** *ip-address* **Example:** `Device(config-router)# network 10.1.1.0`	Associates a network with a RIP routing process.
Step 5	**neighbor** *ip-address* **Example:** `Device(config-router)# neighbor 10.1.1.2`	Defines a neighboring device with which to exchange routing information.
Step 6	**offset-list** [*access-list-number*	access-list-*name*] {**in**	**out**} *offset* [*interface-type interface-number*]	(Optional) Applies an offset list to routing metrics.

	Command or Action	Purpose
	Example: Device(config-router)# offset-list 98 in 1 Ethernet 1/0	
Step 7	**timers basic** *update invalid holddown flush* [*sleeptime*] **Example:** Device(config-router)# timers basic 1 2 3 4	(Optional) Adjusts routing protocol timers.
Step 8	**end** **Example:** Device(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

Specifying a RIP Version and Enabling Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **version** {**1** | **2**}
5. **exit**
6. **interface** *type number*
7. **ip rip send version** [**1**] [**2**]
8. **ip rip receive version** [**1**] [**2**]
9. **ip rip authentication key-chain** *name-of-chain*
10. **ip rip authentication mode** {**text** | **md5**}
11. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**router rip** **Example:** Device(config)# router rip	Enters router configuration mode.
Step 4	**version** {**1** | **2**} **Example:** Device(config-router)# version 2	Enables the Cisco software to send only RIP Version 2 (RIPv2) packets.
Step 5	**exit** **Example:** Device(config-router)# exit	Exits the router configuration mode and enters the global configuration mode.
Step 6	**interface** *type number* **Example:** Device(config)# interface Ethernet 3/0	Specifies an interface and enters interface configuration mode.
Step 7	**ip rip send version** [**1**] [**2**] **Example:** Device(config-if)# ip rip send version 2	Configures an interface to send only RIPv2 packets.
Step 8	**ip rip receive version** [**1**] [**2**] **Example:** Device(config-if)# ip rip receive version 2	Configures an interface to accept only RIPv2 packets.
Step 9	**ip rip authentication key-chain** *name-of-chain* **Example:** Device(config-if)# ip rip authentication key-chain chainname	Enables RIP authentication.
Step 10	**ip rip authentication mode** {**text** | **md5**} **Example:** Device(config-if)# ip rip authentication mode md5	Configures the interface to use message digest algorithm 5 (MD5) authentication (or let it default to plain-text authentication).
Step 11	**end** **Example:** Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Summarizing RIP Routes

RIP Version 2 supports automatic route summarization by default. The software summarizes subprefixes to the classful network boundary when classful network boundaries are crossed. If you have disconnected subnets, disable automatic route summarization to advertise the subnets. When route summarization is disabled, the software sends subnet and host routing information across classful network boundaries. To disable automatic summarization, use the **no auto-summary** command in router configuration mode.

✎

Note   Supernet advertisement (advertising any network prefix less than its classful major network) is not allowed in RIP route summarization, other than advertising a supernet learned in the routing tables. Supernets learned on any interface that is subject to configuration are still learned. For example, the following summarization is invalid: (invalid supernet summarization)

```
Router(config)# interface Ethernet 1
Router(config-if)# ip summary-address rip 10.0.0.0 252.0.0.0
.
.
>
```

SUMMARY STEPS

1. **enable**
2. **configure   terminal**
3. **interface**   *type number*
4. **ip summary-address rip**   *ip-address network-mask*
5. **exit**
6. **router rip**
7. **no auto-summary**
8. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters the interface configuration mode.

Command or Action	Purpose
Router(config)# interface Ethernet 3/0	
Step 4 **ip summary-address rip** *ip-address network-mask* **Example:** Router(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0	Specifies the IP address and network mask that identify the routes to be summarized.
Step 5 **exit** **Example:** Router(config-if)# exit	Exits the interface configuration mode.
Step 6 **router rip** **Example:** Router(config)# router rip	Enters the router configuration mode.
Step 7 **no auto-summary** **Example:** Router(config-router)# no auto-summary	Used in router configuration mode, disables automatic summarization.
Step 8 **end** **Example:** Router(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

Enabling or Disabling Split Horizon

To enable or disable split horizon, use the following commands in interface configuration mode, as needed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip split-horizon**
5. **no ip split-horizon**
6. **end**

DETAILED STEPS

Command or Action	Purpose
Step 1 **enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Router> enable	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** Router(config)# interface Ethernet 3/0	Enters interface configuration mode.
Step 4	**ip split-horizon** **Example:** Router(config-if)# ip split-horizon	Enables split horizon.
Step 5	**no ip split-horizon** **Example:** Router(config-if)# no ip split-horizon	Disables split horizon.
Step 6	**end** **Example:** Router(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling the Validation of Source IP Addresses

Perform this task to disable the default function that validates the source IP addresses of incoming routing updates.

Note Split horizon for Frame Relay and SMDS encapsulation is disabled by default. Split horizon is not disabled by default for interfaces using any of the X.25 encapsulations. For all other encapsulations, split horizon is enabled by default.

In general, changing the state of the default is not recommended unless you are certain that your application requires making a change in order to advertise routes properly. Remember that if split horizon is disabled on a serial interface (and that interface is attached to a packet-switched network), you *must* disable split horizon for all routers in any relevant multicast groups on that network.

Note Summarized network will not be advertised when split horizon is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip split-horizon**
5. **exit**
6. **router rip**
7. **no validate-update-source**
8. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 3/0`	Enters interface configuration mode.
Step 4	**ip split-horizon** **Example:** `Router(config-if)# ip split-horizon`	Enables split horizon.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 6	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode.
Step 7	**no validate-update-source** **Example:** `Router(config-router)# no validate-update-source`	Disables the validation of the source IP address of incoming RIP routing updates.

	Command or Action	Purpose
Step 8	**end** **Example:** Router(config-router)# end	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Interpacket Delay

Perform this to configure interpacket delay.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **exit**
5. **router rip**
6. **output-delay** *milliseconds*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** Router(config)# interface Ethernet 3/0	Enters interface configuration mode.
Step 4	**exit** **Example:** Router(config-if)# exit	Exits interface configuration mode.
Step 5	**router rip** **Example:**	Enters router configuration mode.

	Command or Action	Purpose
	`Router(config)# router rip`	
Step 6	**output-delay** *milliseconds* **Example:** `Router(config-router)# output-delay 8`	Configures interpacket delay for outbound RIP updates.
Step 7	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Optimizing RIP over WAN

There are two problems when RIP is not optimized:

- Periodic broadcasting by RIP generally prevents WAN circuits from being closed.

- Even on fixed, point-to-point links, the overhead of periodic RIP transmissions could seriously interrupt normal data transfer because of the quantity of information that passes through the line every 30 seconds.

To overcome these limitations, triggered extensions to RIP cause RIP to send information on the WAN only when there has been an update to the routing database. Periodic update packets are suppressed over the interface on which this feature is enabled. RIP routing traffic is reduced on point-to-point, serial interfaces. Therefore, you can save money on an on-demand circuit for which you are charged for usage. Triggered extensions to RIP partially support RFC 2091, *Triggered Extensions to RIP to Support Demand Circuits* .

Perform the following task to enable triggered extensions to RIP and to display the contents of the RIP private database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *controller-number*
4. **ip rip triggered**
5. **end**
6. **show ip rip database** [*prefix mask*]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**interface serial** *controller-number* **Example:** Router(config)# interface serial3/0	Configures a serial interface.
Step 4	**ip rip triggered** **Example:** Router(config-if)# ip rip triggered	Enables triggered extensions to RIP.
Step 5	**end** **Example:** Router(config-if)# end	Returns to privileged EXEC mode.
Step 6	**show ip rip database** [*prefix mask*] **Example:** Router# show ip rip database	Displays the contents of the RIP private database.

Configuring IP-RIP Delay Start for Routers Connected by a Frame Relay Network

The tasks in this section explain how to configure a router to use the IP-RIP Delay Start feature on a Frame Relay interface.

Timesaver Cisco routers allow an MD5-authenticated RIPv2 neighbor session to start when the sequence number of the first MD5 packet received from the other router is greater than 0. If you are using only Cisco routers in your network, you do not need to use the IP-RIP Delay Start feature.

Prerequisites

Your router must be running Cisco IOS Release 12.4(12) or a later release.

Note The IP-RIP Delay Start feature is supported over other interface types such as Fast Ethernet and Gigabit Ethernet. If your Cisco router cannot establish RIPv2 neighbor sessions using MD5 authentication with a non-Cisco device, the IP-RIP Delay Start feature might resolve the problem.

Restrictions

The IP-RIP Delay Start feature is required only when your Cisco router is configured to establish a RIPv2 neighbor relationship with a non-Cisco device and you want to use MD5 neighbor authentication.

Configuring RIPv2

This required task configures RIPv2 on the router.

This task provides instructions for only one of the many possible permutations for configuring RIPv2 on your router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **network** *ip-network*
5. **version** {**1** | **2**}
6. [**no**] **auto-summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router rip** **Example:** `Router(config)# router rip`	Enables a RIP routing process, which places you in router configuration mode.
Step 4	**network** *ip-network* **Example:** `Router(config-router)# network 192.168.0.0`	Associates a network with a RIP routing process.
Step 5	**version** {**1**	**2**} **Example:** `Router (config-router)# version 2`	Configures the software to receive and send only RIP Version 1 or only RIP Version 2 packets.

	Command or Action	Purpose
Step 6	[**no**] **auto-summary** **Example:** `Router(config-router)# no auto-summary`	Disables or restores the default behavior of automatic summarization of subnet routes into network-level routes.

Configuring Frame Relay on a Serial Subinterface

This required task configures a serial subinterface for Frame Relay.

✎

Note This task provides instructions for only one of the many possible permutations for configuring Frame Relay on a subinterface. For more information about and instructions for configuring Frame Relay, see the Configuring Frame Relay part of the *Cisco IOS Wide-Area Networking Configuration Guide* .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address**
5. **encapsulation frame-relay [mfr number | ietf]**
6. **frame-relay lmi-type {cisco | ansi | q933a}**
7. **exit**
8. **interface** *type number/subinterface-number* {**point-to-point** | **multipoint**}
9. **frame-relay interface-dlci** *dlci* [**ietf** | **cisco**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface serial3/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	**Purpose**
Step 4	**no ip address** **Example:** `Router(config-if)# no ip address`	Removes a previously configured IP address from the interface.
Step 5	**encapsulation frame-relay [mfr number | ietf]** **Example:** `Router(config-if)# encapsulation frame-relay ietf`	Specifies the type of Frame Relay encapsulation for the interface.
Step 6	**frame-relay lmi-type {cisco | ansi | q933a}** **Example:** `Router(config-if)# frame-relay lmi-type ansi`	Specifies the type of Frame Relay local management interface (LMI) for the interface.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 8	**interface** *type number/subinterface-number* **{point-to-point | multipoint}** **Example:** `Router(config)# interface serial3/0.1 point-to-point`	Specifies a subinterface and the connection type for the subinterface and enters subinterface configuration mode.
Step 9	**frame-relay interface-dlci** *dlci* **[ietf | cisco]** **Example:** `Router(config-subif)# frame-relay interface-dlci 100 ietf`	Assigns a data-link connection identifier (DLCI) to a Frame Relay subinterface.

Configuring IP with MD5 Authentication for RIPv2 and IP-RIP Delay on a Frame Relay Subinterface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *number*
5. **key-string** *string*
6. **exit**
7. **exit**
8. **interface** *type number*
9. **no cdp enable**

10. **ip address** *ip-address subnet-mask*
11. **ip rip authentication mode** {**text** | **md5**}
12. **ip rip authentication key-chain** *name-of-chain*
13. **ip rip initial-delay** *delay*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** Device(config)# key chain rip-md5	Specifies the name of a key chain and enters key chain configuration mode.
Step 4	**key** *number* **Example:** Device(config-keychain)# key 123456	Specifies the key identifier and enters key chain key configuration mode. The range is from 0 to 2147483647.
Step 5	**key-string** *string* **Example:** Device(config-keychain-key)# key-string abcde	Configures the key string.
Step 6	**exit** **Example:** Device(config-keychain-key)# exit	Exits key chain key configuration mode.
Step 7	**exit** **Example:** Device(config-keychain)# exit	Exits key chain configuration mode.
Step 8	**interface** *type* *number* **Example:** Device(config)# interface serial 3/0.1	Specifies a subinterface and enters subinterface configuration mode.

	Command or Action	Purpose
Step 9	**no cdp enable** **Example:** `Device(config-subif)# no cdp enable`	Disables Cisco Discovery Protocol options on the interface. **Note** — Cisco Discovery Protocol is not supported by non-Cisco devices; and the IP-RIP Delay Start feature is required only when you are connecting to a non-Cisco device. Therefore, you should disable Cisco Discovery Protocol on any interfaces on which you want to configure the IP-RIP Delay Start feature.
Step 10	**ip address** *ip-address subnet-mask* **Example:** `Device(config-subif)# ip address 172.16.10.1 255.255.255.0`	Configures an IP address for the Frame Relay subinterface.
Step 11	**ip rip authentication mode** {**text** | **md5**} **Example:** `Device(config-subif)# ip rip authentication mode md5`	Specifies the mode for RIPv2 authentication.
Step 12	**ip rip authentication key-chain** *name-of-chain* **Example:** `Device (config-subif)# ip rip authentication key-chain rip-md5`	Specifies a previously configured key chain for Routing Information Protocol Version (RIPv2) message digest algorithm 5 (MD5) authentication.
Step 13	**ip rip initial-delay** *delay* **Example:** `Device(config-subif)# ip rip initial-delay 45`	Configures the IP-RIP Delay Start feature on the interface. The device will delay sending the first MD5 authentication packet to the RIPv2 neighbor for the number of seconds specified by the *delay* argument. The range is from 0 to 1800.
Step 14	**end** **Example:** `Device(config-subif)# end`	Exits the subinterface configuration mode and returns to privileged EXEC mode.

Configuration Examples for RIP

Route Summarization Example

The following example shows how the **ip summary-address rip** router configuration command can be used to configure summarization on an interface. In this example, the subnets 10.1.3.0/25, 10.1.3.128/25, 10.2.1.0/24,

10.2.2.0/24, 10.1.2.0/24 and 10.1.1.0/24 can be summarized as shown below while sending the updates over an interface.

```
Router(config)#interface GigabitEthernet 0/2
Router(config-if)#ip summary-address rip 10.1.0.0 255.255.0.0
Router(config-if)#ip summary-address rip 10.2.0.0 255.255.0.0
Router(config-if)#ip summary-address rip 10.3.0.0 255.255.0.0
```

Split Horizon Examples

Two examples of configuring split horizon are provided.

Example 1

The following configuration shows a simple example of disabling split horizon on a serial link. In this example, the serial link is connected to an X.25 network.

```
Router(config)# interface Serial 0
Router(config-if)# encapsulation x25

Router(config-if)# no ip split-horizon
```

Example 2

In the next example, the figure below illustrates a typical situation in which the **no ip split-horizon** interface configuration command would be useful. This figure depicts two IP subnets that are both accessible via a serial interface on Router C (connected to a Frame Relay network). In this example, the serial interface on Router C accommodates one of the subnets via the assignment of a secondary IP address.

The Ethernet interfaces for Router A, Router B, and Router C (connected to IP networks 10.13.50.0, 10.155.120.0, and 10.20.40.0, respectively all have split horizon enabled by default, while the serial interfaces connected to networks 172.16.1.0 and 192.168.1.0 all have split horizon disabled with the **no ip split-horizon** command.The figure below shows the topology and interfaces.

In this example, split horizon is disabled on all serial interfaces. Split horizon must be disabled on Router C in order for network 172.16.0.0 to be advertised into network 192.168.0.0 and vice versa. These subnets overlap at Router C, interface S0. If split horizon were enabled on serial interface S0, it would not advertise a route back into the Frame Relay network for either of these networks.

Configuration for Router A

```
interface ethernet 1
 ip address 10.13.50.1
!
interface serial 1
 ip address 172.16.2.2
 encapsulation frame-relay
 no ip split-horizon
```

Configuration for Router B

```
interface ethernet 2
 ip address 10.155.120.1
```

```
!
interface serial 2
 ip address 192.168.1.2
 encapsulation frame-relay
 no ip split-horizon
```

Configuration for Router C

```
interface ethernet 0
 ip address 10.20.40.1
!
interface serial 0
 ip address 172.16.1.1
 ip address 192.168.1.1 secondary
 encapsulation frame-relay
 no ip split-horizon
```

Address Family Timers Example

The following example shows how to adjust individual address family timers. Note that the address family "notusingtimers" will use the system defaults of 30, 180, 180, and 240 even though timer values of 5, 10, 15, and 20 are used under the general RIP configuration. Address family timers are not inherited from the general RIP configuration.

```
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# timers basic 5 10 15 20
Router(config-router)# redistribute connected
Router(config-router)# network 10.5.0.0
Router(config-router)# default-metric 10
Router(config-router)# no auto-summary
Router(config-router)#
Router(config-router)# address-family ipv4 vrf abc
Router(config-router-af)# timers basic 10 20 20 20
Router(config-router-af)# redistribute connected
Router(config-router-af)# network 10.0.0.0
Router(config-router-af)# default-metric 5
Router(config-router-af)# no auto-summary
Router(config-router-af)# version 2
Router(config-router-af)# exit-address-family
Router(config-router)#
Router(config-router)# address-family ipv4 vrf xyz
Router(config-router-af)# timers basic 20 40 60 80
Router(config-router-af)# redistribute connected
Router(config-router-af)# network 10.20.0.0
Router(config-router-af)# default-metric 2
Router(config-router-af)# no auto-summary
Router(config-router-af)# version 2
Router(config-router-af)# exit-address-family
Router(config-router)#
Router(config-router)# address-family ipv4 vrf notusingtimers

Router(config-router-af)# redistribute connected
Router(config-router-af)# network 10.20.0.0
Router(config-router-af)# default-metric 2
Router(config-router-af)# no auto-summary
Router(config-router-af)# version 2
Router(config-router-af)# exit-address-family
Router(config-router)#
```

Example: IP-RIP Delay Start on a Frame Relay Interface

Additional References

The following sections provide references related to configuring Routing Information Protocol.

Related Documents

Related Topic	Document Title
Protocol-independent features, filtering RIP information, key management (available in RIP Version 2), and VLSM	*Configuring IP Routing Protocol-Independent Features*
IPv6 Routing: RIP for IPv6	*Cisco IOS IP Routing: RIP Configuration Guide*
RIP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: RIP Command Reference*
Configuring Frame Relay	*Cisco IOS Wide-Area Networking Configuration Guide*

Standards

Standard	Title
None	--

MIBs

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

RFCs

RFC	Title
RFC 1058	*Routing Information Protocol*
RFC 2082	RIP-2 MD5 Authentication
RFC 2091	*Triggered Extensions to RIP to Support Demand Circuits*
RFC 2453	RIP version 2

Technical Assistance

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

Feature Information for Configuring RIP

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

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

Table 3: Feature Information for Configuring Routing Information Protocol

Feature Name	Releases	Feature Information
IP-RIP Delay Start	12.4(12), 15.0(1)M, 12.2(33)SRE, 15.0(1)SY	The IP-RIP Delay Start feature is used on Cisco routers to delay the initiation of RIPv2 neighbor sessions until the network connectivity between the neighbor routers is fully operational, thereby ensuring that the sequence number of the first MD5 packet that the router sends to the non-Cisco neighbor router is 0. The default behavior for a router configured to establish RIPv2 neighbor sessions with a neighbor router using MD5 authentication is to start sending MD5 packets when the physical interface is up. The following commands were introduced or modified: **ip rip initial-delay**.

Feature Name	Releases	Feature Information
IP Summary Address for RIPv2	12.0(7)T 12.1(3)T 12.1(14) 12.2(2)T 12.2(27)SBB 15.0(1)M 12.2(33)SRE 15.0S	The IP Summary Adddress for RIPv2 feature introduced the ability to summarize routes. Summarizing routes in RIP Version 2 improves scalability and efficiency in large networks. Summarizing IP addresses means that there is no entry for child routes (routes that are created for any combination of the individual IP addresses contained within a summary address) in the RIP routing table, reducing the size of the table and allowing the router to handle more routes. The following commands were introduced or modified by this feature: **ip summary-address rip**.
Routing Information Protocol	12.2(27)SBB 15.0(1)M 12.2(33)SRE 15.0S	Routing Information Protocol (RIP) is a commonly used routing protocol in small to medium TCP/IP networks. It is a stable protocol that uses a distance-vector algorithm to calculate routes.
Triggered RIP	12.0(1)T 15.0(1)M 12.2(33)SRE 15.0S	Triggered RIP was introduced to overcome constant RIP updates over expensive circuit-based WAN links. Triggered extensions to RIP cause RIP to send information on the WAN only when there has been an update to the routing database. Periodic update packets are suppressed over the interface on which this feature is enabled. RIP routing traffic is reduced on point-to-point, serial interfaces. The following commands were introduced or modified: **ip rip triggered**, **show ip rip database**.

Glossary

address family --A group of network protocols that share a common format of network address. Address families are defined by RFC 1700.

IS-IS --Intermediate System-to-Intermediate System. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, where routers exchange routing information based on a single metric, to determine network topology.

RIP --Routing Information Protocol.RIP is a dynamic routing protocol used in local and wide area networks.

VRF --VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

BFD for RIPv2 Support

The BFD for RIPv2 Support feature is used to facilitate an alternate path selection when a neighboring router is inactive.

Routing Information Protocol (RIP) uses the timeout of prefixes of a particular neighbor to identify if a neighbor is inactive. By default, the timeout is 180 seconds; that is, although the next-hop router is inactive, the RIP router will still broadcast prefixes for up to 180 seconds.

Bidirectional Forward Detection (BFD) is a protocol that provides subsecond failure detection using a single, common standardized mechanism that is independent of media and routing protocols.

Finding Feature Information

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

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

Prerequisites for BFD for RIPv2 Support

BFD is independent of RIPv2 and must be enabled and functional on the router.

How to Configure BFD for RIPv2 Support Feature

Configuring BFD on RIPv2 Neighbors

Perform this task to configure BFD on RIPv2 neighbors:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**
4. **bfd all-interfaces**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router rip** **Example:** `Router(config)# router rip`	Configures the RIP routing process and enters router configuration mode.
Step 4	**bfd all-interfaces** **Example:** `Router(config-router)# bfd all-interfaces`	Enables BFD on all interfaces associated with the routing process. • RIPv2 registers with BFD and creates sessions for the neighbor when RIP updates are received. New neighbors are automatically enabled for BFD when the update packets are received. **Note** Alternatively, you can use the **neighbor** *ip-address* **bfd**command to enable BFD for a specific RIP neighbor.
Step 5	**end** **Example:**	Exits router configuration mode and returns to global configuration mode.

Command or Action	Purpose
Router(config-router)# end	

Configuration Example for BFD for RIPv2 Support Feature

Example Configuring BFD for a RIPv2 Neighbor

The following example shows how to configure BFD for all interfaces associated with a RIPv2 neighbor:

```
!
interface GigabitEthernet 0/0/0
 ip address 10.10.10.1 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 5
end
!
interface GigabitEthernet 0/0/1
 ip address 10.10.20.1 255.255.255.0
 bfd interval 50 min_rx 50 multiplier 5
end
!
router rip
 version 2
 redistribute connected
 network 10.0.0.0
 neighbor 10.10.20.2 bfd
 bfd all-interfaces
 no auto-summary
!
```

Additional References

Related Documents

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

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	--

RFCs

RFC	Title
None	--

Technical Assistance

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

Feature Information for BFD for RIPv2 Support

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

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

Table 4: Feature Information for BFD for RIPv2 Support

Feature Name	Releases	Feature Information
BFD for RIPv2 Support	Cisco IOS XE Release 3.3	The BFD for RIPv2 Support feature is used to facilitate alternate path selection when a neighboring router is inactive. The following commands were introduced or modified: **bfd all-interfaces, debug ip rip bfd events, neighbor (RIP),**and **show ip rip neighbor**.

IPv6: RIPng VRF-Aware Support

The IPv6: RIPng VRF-Aware Support feature uses separate routing tables for every provider edge-customer edge (PE-CE) scenario, thus allowing improved route protection, modularity, and a potential reduction in the size of the routing table.

Finding Feature Information

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

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

Information About IPv6: RIPng VRF-Aware Support

IPv6 Routing: RIP for IPv6

IPv6 Routing Information Protocol (RIP) functions the same and offers the same benefits as IPv4 RIP. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes and the use of the all-RIP-devices multicast group address, FF02::9, as the destination address for RIP update messages.

IPv6: RIPng VRF-Aware Support

When not in Virtual Routing and Forwarding (VRF) mode, every IPv6 Routing Information Protocol (RIP)—also know as RIP Next Generation (RIPng)—process and the configuration associated with it, keeps

all the routes in the same routing table. In other routing protocols, it is often required to keep the protocol-related routes stored in separate routing tables.

The IPv6: RIPng VRF-Aware Support feature enables isolation, modularity, and potential performance improvement by reducing the number of routes stored in a single routing table. It also allows a network administrator to create different RIP routing tables and share the same protocol configuration stored in a single RIP protocol configuration block.

How to Configure IPv6: RIPng VRF-Aware Support

Configuring IPv6: RIPng VRF-Aware Support

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **vrf definition** *vrf-name*
5. **address-family ipv6**
6. **exit**
7. **exit**
8. **ipv6 rip vrf-mode enable**
9. **ipv6 router rip** *rip-process-name*
10. **exit**
11. **interface** *type number*
12. **vrf forwarding** *vrf-name*
13. **ipv6 enable**
14. **ipv6 rip** *rip-process-name* **enable**
15. **end**
16. **debug ipv6 rip vrf** *vrf-name*
17. **show ipv6 rip vrf** *vrf-name* **next-hops**
18. **show ipv6 rip vrf** *vrf-name* **database**

DETAILED STEPS

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

	Command or Action	**Purpose**
Step 3	**ipv6 unicast-routing** **Example:** `Device (config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Configures a virtual routing and forwarding (VRF) routing table instance and enters VRF configuration mode.
Step 5	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Enters VRF address family configuration mode and enables IPv6 address prefixes.
Step 6	**exit** **Example:** `Device(config-vrf-af)# exit`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 7	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 8	**ipv6 rip vrf-mode enable** **Example:** `Device (config)# ipv6 rip vrf-mode enable`	Enables VRF support for IPv6 RIP routing and enters RTR entry configuration mode.
Step 9	**ipv6 router rip** *rip-process-name* **Example:** `Device (config)# ipv6 router rip myrip`	Creates an IPv6 Routing Information Protocol (RIP) routing process instance.
Step 10	**exit** **Example:** `Device (config-rtr)# exit`	Exits RTR entry configuration mode and returns to global configuration mode.
Step 11	**interface** *type* *number* **Example:** `Device (config)# interface Ethernet 0/0`	Specifies the interface type and number and enters interface configuration mode.
Step 12	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Binds the interface to the specified VRF routing instance table and removes all the Layer 3 interface configuration that is available when the command is entered.
Step 13	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface.

	Command or Action	**Purpose**
Step 14	**ipv6 rip** *rip-process-name* **enable** **Example:** `Device(config-if)# ipv6 rip myrip enable`	Enables an IPv6 RIP routing process on the interface.
Step 15	**end** **Example:** `Device (config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 16	**debug ipv6 rip vrf** *vrf-name* **Example:** `Device# debug ipv6 rip vrf vrf1`	Displays debugging information related to VRF support for the specified IPv6 RIP VRF routing table instance.
Step 17	**show ipv6 rip vrf** *vrf-name* **next-hops** **Example:** `Device# show ipv6 rip vrf vrf1 next-hops`	Displays the next hops in the specified VRF RIPng routing table.
Step 18	**show ipv6 rip vrf** *vrf-name* **database** **Example:** `Device# show ipv6 rip vrf vrf1 database`	Displays the associated RIP local routing information base (RIB).

Configuration Examples for IPv6: RIPng VRF-Aware Support

Example: Configuring IPv6: RIPng VRF-Aware Support

```
Device> enable
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# vrf definition vrf1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# ipv6 rip vrf-mode enable
Device(config)# ipv6 router rip myrip
Device(config-rtr)# exit
Device(config)# interface Ethernet 0/0
Device(config-if)# vrf forwarding vrf1
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 rip myrip enable
Device(config-if)# end
```

Example: Verifying IPv6: RIPng VRF-Aware Support

```
Device> debug ipv6 rip vrf vrf1

RIP Routing Protocol debugging is on for vrf vrf1
Sending:
*Mar 15 11:23:08.508: RIPng: Sending multicast update on Ethernet0/0 for vrf for vrf vrf1
*Mar 15 11:23:08.508: src=2001:DB8:0:1:FFFF:1234::5
*Mar 15 11:23:08.508: dst=2001:DB8:0:1::1 (Ethernet0/0)
*Mar 15 11:23:08.508: sport=521, dport=521, length=52
*Mar 15 11:23:08.508: command=2, version=1, mbz=0, #rte=2
*Mar 15 11:23:08.508: tag=0, metric=1, prefix=6000::/64
*Mar 15 11:23:08.508: tag=0, metric=1, prefix=2000::/64
*Mar 15 11:23:08.508: RIPng: Packet waiting
*Mar 15 11:23:08.508: RIPng: Process vrf received own response on Loopback1
Receiving
*Mar 15 11:23:20.316: RIPng: Packet waiting
*Mar 15 11:23:20.316: RIPng: response received from FE80::A8BB:CCFF:FE00:7C00 on Ethernet0/0
for vrf
*Mar 15 11:23:20.316: src=2001:DB8:0:1:FFFF:1234::4 (Ethernet0/0)
*Mar 15 11:23:20.316: dst=2001:DB8::1
*Mar 15 11:23:20.316: sport=521, dport=521, length=32
*Mar 15 11:23:20.316: command=2, version=1, mbz=0, #rte=1
*Mar 15 11:23:20.316: tag=0, metric=1, prefix=AAAA::/64


Device> show ipv6 rip vrf vrf1 database

RIP VRF "vrf1", local RIB
FE80::A8BB:CCFF:FE00:7C00/Ethernet0/0 [1 paths]


Device> show ipv6 rip vrf vrf1 next-hops

RIP VRF "vrf1", Next Hops
 AAAA::/64, metric 2, installed
Ethernet0/0/FE80::A8BB:CCFF:FE00:7C00, expires in 177 secs
```

Additional References for IPv6: RIPng VRF-Aware Support

Related Documents

Related Topic	Document Title
IP Routing: RIP commands	Cisco IOS IP Routing: RIP Command Reference
IPv6 Routing: RIP for IPv6	*Cisco IOS IP Routing: RIP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2080	*RIPng for IPv6*

Technical Assistance

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

Feature Information for IPv6: RIPng VRF-Aware Support

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

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

Table 5: IPv6: RIPng VRF-Aware Support

Feature Name	Releases	Feature Information
IPv6: RIPng VRF-Aware Support	15.3(3)M 15.2(1)SY	When not virtual routing and forwarding (VRF) aware, IPv6 Routing Information Protocol (RIP), also known as RIP Next Generation (RIPng), works only with routes that are available in the default global routing table. When operating in VRF mode, RIPng, creates a separate routing table for each VRF instance. The IPv6: RIPng VRF-Aware Support feature enables the availability of separate routing tables for every provider edge-customer edge (PE-CE) scenario, thus allowing improved route protection, modularity, and a potential reduction in the size of the routing table. The following commands were introduced or modified: **clear ipv6 rip**, **debug ipv6 rip** , **ipv6 rip vrf-mode enable**, and **show ipv6 rip**.