

Configure RIPng

- RIPng, on page 1
- Prerequisites for RIPng, on page 3
- Guidelines and limitations for RIPng, on page 3
- Default settings for RIPng parameters, on page 4
- Configure RIPng, on page 4
- Verify the RIPng configuration, on page 13
- Display RIPng statistics, on page 14
- Configuration Examples for RIPng, on page 14
- Related topics, on page 14

RIPng

RIPng

RIPng uses User Datagram Protocol (UDP) data packets to exchange routing information in small internetworks.

RIPng supports IPv6 and uses the following two message types:

- Request—Sent to the multicast address FF02::9 to request route updates from other RIPng-enabled routers.
- Response—Sent every 30 seconds by default (see the Verify the RIPng configuration, on page 13 section). The router also sends response messages after it receives a request message. The response message contains the entire RIPng route table. RIPng sends multiple response packets for a request if the RIPng routing table cannot fit in one response packet.

RIPng uses a hop count for the routing metric. The hop count is the number of routers that a packet can traverse before reaching its destination. A directly connected network has a metric of 1. An unreachable network has a metric of 16. This small range of metrics makes RIPng an unsuitable routing protocol for large networks.

Split Horizon

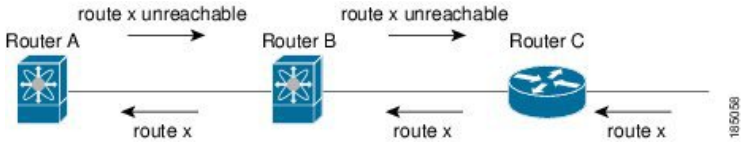
You can use split horizon to ensure that RIPng never advertises a route out of the interface where it was learned.

Split horizon is a method that controls the sending of RIPng update and query packets. When you enable split horizon on an interface, Cisco NX-OS does not send update packets for destinations that were learned from this interface. Controlling update packets in this manner reduces the possibility of routing loops.

You can use split horizon with poison reverse to configure an interface to advertise routes learned by RIPng as unreachable over the interface that learned the routes.

The following figure shows a sample RIPng network with split horizon and poison reverse enabled.

Figure 1: RIPng with Split Horizon Poison Reverse



Router C learns about route X and advertises that route to Router B. Router B in turn advertises route X to Router A but sends a route X unreachable update back to Router C.

By default, split horizon is enabled on all interfaces.

Route filter

You can configure a route policy on an RIPng-enabled interface to filter the RIPng updates. Cisco NX-OS updates the route table with only those routes that the route policy allows.

Load balance

You can use load balancing to allow a router to distribute traffic over all the router network ports that are the same distance from the destination address. Load balancing increases the usage of network segments and increases effective network bandwidth.

Cisco NX-OS supports the Equal Cost Multiple Paths (ECMP) feature with up to 16 equal-cost paths in the RIPng route table and the unicast RIB. You can configure RIPng to load balance traffic across some or all of those paths.

Default information origination and generation

Cisco NX-OS supports default-information origination and generation for RIPng IPv6.

To generate a default route into the Routing Information Protocol (RIP), use the default-information originate command in router address-family configuration mode. To disable this feature, use the **no** form of this command.

default-information originate [**always**] [**route-map** *map-name*]

no default-information originate

Note Use the always keyword to generate the default route if the route is not present in the RIP routing information base, that is the RIP internal RIB. Use the route-map keyword along with the map-name variable to generate the default route only if the route is permitted by the route map. The map name is any alphanumerical string up to 63 characters. Use the originate to send the default route along with regular updates.

The following example shows how to originate a default route to all routes that pass the condition route map.

switch(config)# **router rip Enterprise**

switch(config-router)# **address-family ipv6 unicast**

switch(config-router-af)# **default-information originate route-map Condition**

High availability for RIPng

Cisco NX-OS supports stateless restarts for RIPng. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration, and RIPng immediately sends request packets to repopulate its routing table.

Virtualization for RIPng

Cisco NX-OS supports multiple instances of the RIPng protocol that run on the same system. RIPng supports virtual routing and forwarding (VRF) instances.

Prerequisites for RIPng

RIPng has these prerequisites:

- You must enable RIPng (see the Enable RIPng, on page 4 section).

Guidelines and limitations for RIPng

RIPng has the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.2(3)F, RIPng feature is introduced to support IPv6 on Cisco Nexus 9300 and 9500 series platform switches.
- Names in the prefix-list are case-insensitive. We recommend using unique names. Do not use the same name by modifying upper-case and lower-case characters. For example, CTCPrimaryNetworks and CtcPrimaryNetworks are not two different entries.
- Cisco NX-OS does not support RIPv1. If Cisco NX-OS receives an RIPv1 packet, it logs a message and drops the packet.
- Cisco NX-OS does not establish adjacencies with RIPv1 routers.

Default settings for RIPng parameters

The table lists the default settings for RIPng parameters.

Default RIPng Parameters

Parameters	Default
Maximum paths for load balancing	16
RIPng feature	Disabled
Split horizon	Enabled

Configure RIPng

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enable RIPng

You must enable RIPng before you can configure RIPng.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
      switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **feature rip**

Example:

```
switch(config)# feature rip
```

Enables the RIPng feature.

Step 3 (Optional) **show feature**

Example:

```
switch(config)# show feature
```

Displays enabled and disabled features.

Step 4 (Optional) **copy running-config startup-config**

Example:

```
switch(config)# copy running-config
      startup-config
```

Saves this configuration change.

Create an RIPng instance

You can create an RIPng instance and configure the address family for that instance.

Before you begin

You must enable RIPng (see the Enable RIPng, on page 4 section).

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
      switch(config)#
```

Enters global configuration mode.

Step 2 **[no] router rip** *instance-tag*

Example:

```
switch(config)# router RIP Enterprise
      switch(config-router)#
```

Creates a new RIPng instance with the configured *instance-tag*.

Step 3 **address-family ipv6 unicast**

Example:

```
switch(config-router)# address-family ipv6 unicast
      switch(config-router-af)#
```

Configures the address family for this RIPng instance and enters address-family configuration mode.

Step 4 (Optional) **show ipv6 rip** [**instance** *instance-tag*] [**vrf** *vrf-name*]

Example:

```
switch(config-router-af)# show ipv6 rip
```

Displays a summary of RIPng information for all RIPng instances.

Step 5 (Optional) **distance** *value*

Example:

```
switch(config-router-af)# distance 30
```

Sets the administrative distance for RIPng. The range is from 1 to 255. The default is 120. See the Administrative Distance section.

Step 6 (Optional) **maximum-paths** *number*

Example:

```
switch(config-router-af)# maximum-paths 6
```

Configures the maximum number of equal-cost paths that RIPng maintains in the route table. The range is from 1 to 64. The default is 16.

Step 7 (Optional) **copy running-config startup-config**

Example:

```
switch(config-router-af)# copy running-config
      startup-config
```

Saves this configuration change.

This example shows how to create an RIPng instance for IPv6 and set the number of equal-cost paths for load balancing:

```
switch# configure terminal
    switch(config)# router rip Enterprise
    switch(config-router)# address-family ipv6 unicast
    switch(config-router-af)# max-paths 10
    switch(config-router-af)# copy running-config startup-config
```

Restart an RIPng instance

You can restart an RIPng instance and remove all associated neighbors for the instance.

To restart an RIPng instance and remove all associated neighbors, use the following command in global configuration mode:

Procedure

---

restart rip *instance-tag*

Example:

```
switch(config)# restart rip Enterprise
```

Restarts the RIPng instance and removes all neighbors.

Configure RIPng on an interface

Before you begin

You must enable RIPng (see the Enable RIPng, on page 4 section).

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
      switch(config)#
```

Enters global configuration mode.

Step 2 **interface** *interface-type slot/port*

Example:

```
switch(config)# interface ethernet 1/2
      switch(config-if)#
```

Enters interface configuration mode.

Step 3 **ipv6 router rip** *instance-tag*

Example:

```
switch(config-if)# ipv6 router rip Enterprise
```

Associates this interface with an RIPng instance.

Step 4 (Optional) **show ipv6 rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]

Example:

```
switch(config-if)# show ipv6 rip Enterprise ethernet 1/2
```

Displays RIPng information for an interface.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

Saves this configuration change.

This example shows how to add Ethernet 1/2 interface to an RIPng instance:

```
switch# configure terminal
    switch(config)# interface ethernet 1/2
    switch(config-if)# ipv6 router rip Enterprise
    switch(config)# copy running-config startup-config
```

Configure split horizon with poison reverse

You can configure an interface to advertise routes learned by RIPng as unreachable over the interface that learned the routes by enabling poison reverse.

To configure split horizon with poison reverse on an interface, use the following command in interface configuration mode:

Procedure

ipv6 rip poison-reverse

Example:

```
switch(config-if)# ipv6 rip poison-reverse
```

Enables split horizon with poison reverse. Split horizon with poison reverse is disabled by default.

Configure Cisco NX-OS RIPng for compatibility with Cisco IOS RIPng

You can configure Cisco NX-OS RIPng to behave like Cisco IOS RIPng in the way that routes are advertised and processed.

Directly connected routes are treated with cost 1 in Cisco NX-OS RIPng and with cost 0 in Cisco IOS RIPng. When routes are advertised in Cisco NX-OS RIPng, the receiving device adds a minimum cost of +1 to all received routes and installs the routes in its routing table. In Cisco IOS RIPng, this cost increment is done on the sending router, and the receiving router installs the routes without any modification. This difference in behavior can cause issues when both Cisco NX-OS and Cisco IOS devices are working together. You can prevent these compatibility issues by configuring Cisco NX-OS RIPng to advertise and process routes like Cisco IOS RIPng.

Before you begin

You must enable RIPng (see the Enable RIPng, on page 4 section).

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
      switch(config)#
```

Enters global configuration mode.

Step 2 **router rip** *instance-tag*

Example:

```
switch(config)# router rip 100
      switch(config-router)#
```

Creates a new RIPng instance with the configured instance tag. You can enter 100, 201, or up to 20 alphanumeric chapters for the instance tag.

Step 3 [**no**] **metric direct 0**

Example:

```
switch(config-router)# metric direct 0
```

Configures all directly connected routes with cost 0 instead of the default of cost 1 in order to make Cisco NX-OS RIPng compatible with Cisco IOS RIPng in the way that routes are advertised and processed.

Note This command must be configured on all Cisco NX-OS devices that are present in any RIPng network that also contains Cisco IOS devices.

Step 4 (Optional) **show running-config rip**

Example:

```
switch(config-router)# show
      running-config rip
```

Displays the current running RIPng configuration.

Step 5 (Optional) **copy running-config startup-config**

Example:

```
switch(config-router)# copy running-config
      startup-config
```

Saves this configuration change.

This example shows how to disable Cisco NX-OS RIPng compatibility with Cisco IOS RIPng by returning all direct routes from cost 0 to cost 1:

```
switch# configure terminal
    switch(config)# router rip 100
    switch(config-router)# no metric direct 0
    switch(config-router)# show running-config rip
    switch(config-router)# copy running-config startup-config
```

Configure virtualization

You can configure multiple RIPng instances, create multiple VRFs, and use the same or multiple RIPng instances in each VRF. You assign an RIPng interface to a VRF.



Note Configure all other parameters for an interface after you configure the VRF for an interface. Configuring a VRF for an interface deletes all the configurations for that interface.

Before you begin

You must enable RIPng (see the Enable RIPng, on page 4 section).

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
      switch(config)#
```

Enters global configuration mode.

Step 2 **vrf context** *vrf-name*

Example:

```
switch(config)# vrf context RemoteOfficeVRF
      switch(config-vrf)#
```

Creates a new VRF and enters VRF configuration mode.

Step 3 **exit**

Example:

```
switch(config-vrf)# exit
      switch(config)#
```

Exits VRF configuration mode.

Step 4 **router rip** *instance-tag*

Example:

```
switch(config)# router rip Enterprise
      switch(config-router)#
```

Creates a new RIPng instance with the configured instance tag.

Step 5 **vrf** *vrf-name*

Example:

```
switch(config-router)# vrf RemoteOfficeVRF
      switch(config-router-vrf)#
```

Creates a new VRF.

Step 6 (Optional) **address-family ipv6 unicast**

Example:

```
switch(config-router-vrf)# address-family ipv6 unicast
      switch(config-router-vrf-af)#
```

Configures the VRF address family for this RIPng instance.

Step 7 **interface ethernet** *slot*/*port*

Example:

```
switch(config-router-vrf-af)# interface ethernet 1/2
      switch(config-if)#
```

Enters interface configuration mode.

Step 8 **vrf member** *vrf-name*

Example:

```
switch(config-if)# vrf member RemoteOfficeVRF
```

Adds this interface to a VRF.

Step 9 **ipv6 address** *ipv6-prefix*/*length*

Example:

```
switch(config-if)# ipv6 address 1001::1/64
```

Configures an IP address for this interface. You must perform this step after you assign this interface to a VRF.

Step 10 **ipv6 router rip** *instance-tag*

Example:

```
switch(config-if)# ipv6 router rip Enterprise
```

Associates this interface with an RIPng instance.

Step 11 (Optional) **show ipv6 rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*]

Example:

```
switch(config-if)# show ipv6 rip Enterprise ethernet 1/2
```

Displays RIPng information for an interface in a VRF.

Step 12 (Optional) **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

Saves this configuration change.

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
    switch(config)# vrf context RemoteOfficeVRF
    switch(config-vrf)# exit
    switch(config)# router rip Enterprise
    switch(config-router)# vrf RemoteOfficeVRF
    switch(config-router-vrf)# address-family ipv6 unicast
    switch(config-router-vrf-af)# interface ethernet 1/2
    switch(config-if)# vrf member RemoteOfficeVRF
    switch(config-if)# ipv6 address 1001::1/64
    switch(config-if)# ipv6 router rip Enterprise
    switch(config-if)# copy running-config startup-config
```

Tune RIPng

You can tune RIPng to match your network requirements. RIPng uses several timers that determine 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.

Note You must configure the same values for the RIPng timers on all RIPng-enabled routers in your network.

You can use the following optional commands in address-family configuration mode to tune RIPng:

Command	Purpose
timers basic *update timeout holddown garbage-collection* **Example:** `switch(config-router-af)# timers basic 40 120 120 100`	Sets the RIPng timers in seconds. The parameters are as follows: • *update*—The range is from 5 to any positive integer. The default is 30. • *timeout*—The time that Cisco NX-OS waits before declaring a route as invalid. If Cisco NX-OS does not receive route update information for this route before the timeout interval ends, Cisco NX-OS declares the route as invalid. The range is from 1 to any positive integer. The default is 180. • *holddown*—The time during which Cisco NX-OS ignores better route information for an invalid route. The range is from 0 to any positive integer. The default is 180. • *garbage-collection*—The time from when Cisco NX-OS marks a route as invalid until Cisco NX-OS removes the route from the routing table. The range is from 1 to any positive integer. The default is 120.

You can use the following optional commands in interface configuration mode to tune RIPng:

Command	Purpose
ipv6 rip route-filter {**prefix-list** *list-name* | **route-map** *map-name* | [**in** | **out**] **Example:** `switch(config-if)# ipv6 rip route-filter route-map InputMap in`	Specifies a route map to filter incoming or outgoing RIPng updates.

Verify the RIPng configuration

To display the RIPng configuration, perform one of the following tasks.

Command	Purpose
show ipv6 rip instance [*instance-tag*] [**vrf** *vrf-name*]	Displays the status for an instance of RIPng.
show ipv6 rip [**instance** *instance-tag*] **interface** *slot/port* **detail** [**vrf** *vrf-name*]	Displays the RIPng status for an interface.
show ipv6 rip [**instance** *instance-tag*] **neighbor** [*interface-type number*] [**vrf** *vrf-name*]	Displays the RIPng neighbor table.

Command	Purpose
show ipv6 rip [**instance** *instance-tag*] **route** [*ip-prefix/length* [**longer-prefixes** | **shorter-prefixes**]] [**summary**] [**vrf** *vrf-name*]	Displays the RIPng route table.
show running-configuration rip	Displays the current running RIPng configuration.

Display RIPng statistics

To display RIPng statistics, use the following commands.

Command	Purpose
show ipv6 rip [**instance** *instance-tag*] **statistics** *interface-type number* [**vrf** *vrf-name*]	Displays the RIPng statistics.

Use the **clear ipv6 rip statistics** command to clear RIPng statistics.

Configuration Examples for RIPng

The following example shows how to create the Enterprise RIPng instance in a VRF and add Ethernet interface 1/2 to this RIPng instance.

```
router rip Enterprise
address-family ipv6 unicast
distance 33
maximum-paths 8
default-information originate always
timers basic 31 181 181 121
interface ethernet 1/2
ipv6 router rip Enterprise
```

Related topics

See Configuring Route Policy Manager for more information on route maps.