



# Routing Information Protocol

The Routing Information Protocol (RIP) is a distance-vector routing protocol that uses broadcast or multicast User Datagram Protocol (UDP) data packets to exchange routing information. Commonly deployed in small to medium TCP/IP networks, RIP calculates routes based on hop count and sends routing-update messages at regular intervals and upon network topology changes.

- [Feature history for RIP, on page 1](#)
- [RIP in Cisco Catalyst SD-WAN, on page 2](#)
- [Configure RIPv2 using the CLI, on page 5](#)
- [Verification commands for RIPv2 configuration, on page 8](#)
- [Configure RIPng using the CLI, on page 10](#)
- [Verification commands for RIPng configuration, on page 12](#)

## Feature history for RIP

This table describes the developments of this feature, by release.

**Table 1: Feature history**

| Feature Name  | Release Information   | Description   |
|---|---|---|
| RIPv2 support on Cisco IOS XE Catalyst SD-WAN devices | Cisco IOS XE Catalyst SD-WAN Release 17.7.1a<br>Cisco vManage Release 20.7.1<br>Cisco SD-WAN Release 20.7.1 | RIPv2 support on Cisco IOS XE Catalyst SD-WAN devices enables you to configure RIPv2 on these devices. Routers redistribute RIPv2 routes to Overlay Management Protocol (OMP) for advertisement in the Cisco Catalyst SD-WAN overlay, and to Open Shortest Path First version 3 (OSPFv3) for service-side routing |

| Feature Name   | Release Information  | Description   |
|--|--|---|
| RIPng (IPv6) support on Cisco IOS XE Catalyst SD-WAN devices | Cisco IOS XE Catalyst SD-WAN Release 17.10.1a<br>Cisco vManage Release 20.10.1 | RIPng (IPv6) Support on Cisco IOS XE Catalyst SD-WAN devices adds support for IPv6 addresses and prefixes. This feature facilitates redistribution of connect, static, Overlay Management Protocol (OMP), and Open Shortest Path First (OSPF) routes into Routing Information Protocol next generation (RIPng). |

## RIP in Cisco Catalyst SD-WAN

The Routing Information Protocol (RIP) is a distance-vector algorithm-based routing protocol that uses broadcast or multicast User Datagram Protocol (UDP) data packets to exchange routing information. It is commonly used in small to medium TCP/IP networks. Cisco IOS software sends routing information updates every 30 seconds, termed as advertising, at regular intervals and when the network topology changes.

### RIPv2 (RIP for IPv4)

RIP Version 2 (RIPv2) is the Cisco IOS software implementation of RIP for IPv4 networks. Each RIP process maintains a local database containing best-cost RIP routes learned from neighboring RIP-enabled routers. Route redistribution allows routes to be specified by a prefix, using a route map and prefix list.

### RIPng (RIP for IPv6)

Routing Information Protocol next generation (RIPng) is a UDP-based protocol designed for communicating routing information to compute routes through IPv6 networks. RIPng enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes.

### Key Features of RIPv2

The Cisco implementation of RIPv2 supports:

- Plain text and Message Digest Algorithm 5 (MD5) authentication.
- Route summarization.
- Classless Inter-Domain Routing (CIDR).
- Variable-Length Subnet Masks (VLSMs).

If you are sending and receiving RIPv2 packets, we recommend that you enable RIP authentication on an interface because RIPv1 does not support authentication. Plain text authentication is the default authentication in every RIPv2 packet.

### Default Behavior and Configuration (RIPv2)

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, or only RIPv2 packets. The RIP

version that an interface sends can be configured to override this default behavior, and the processing of received packets can also be controlled. RIPv2 is supported on both the service side and transport side.



---

**Note** For network configuration, we recommend that you use Classful IP Network ID Addressing.

---

### **RIPng Redistribution Support**

As an Interior Gateway Protocol (IGP), RIPng supports the redistribution of:

- OMP routes into RIP
- RIP routes into OMP
- RIP routes into OSPFv3
- OSPFv3 routes into RIP
- Static routes into RIP
- RIP routes into static
- Connect routes into RIP
- RIP routes into connect

Each router that implements RIPng requires a routing table containing the following fields:

- The IPv6 prefix of the destination.
- Metric: The total cost of the metric advertised for the address.
- Route Tag: A route attribute that must be advertised and redistributed with the route.
- Next-hop IPv6 address of the destination.
- Various timers associated with the routes.

### **RIPng VRF-aware Support**

When not in Virtual Routing and Forwarding (VRF) mode, every IPv6 RIPng process and its associated configuration maintain all routes in the same routing table. IPv6 RIPng VRF-aware support enhances isolation, modularity, and potential performance by reducing the number of routes stored in a single routing table. It also allows network administrators to create different RIP routing tables and share a single protocol configuration block.

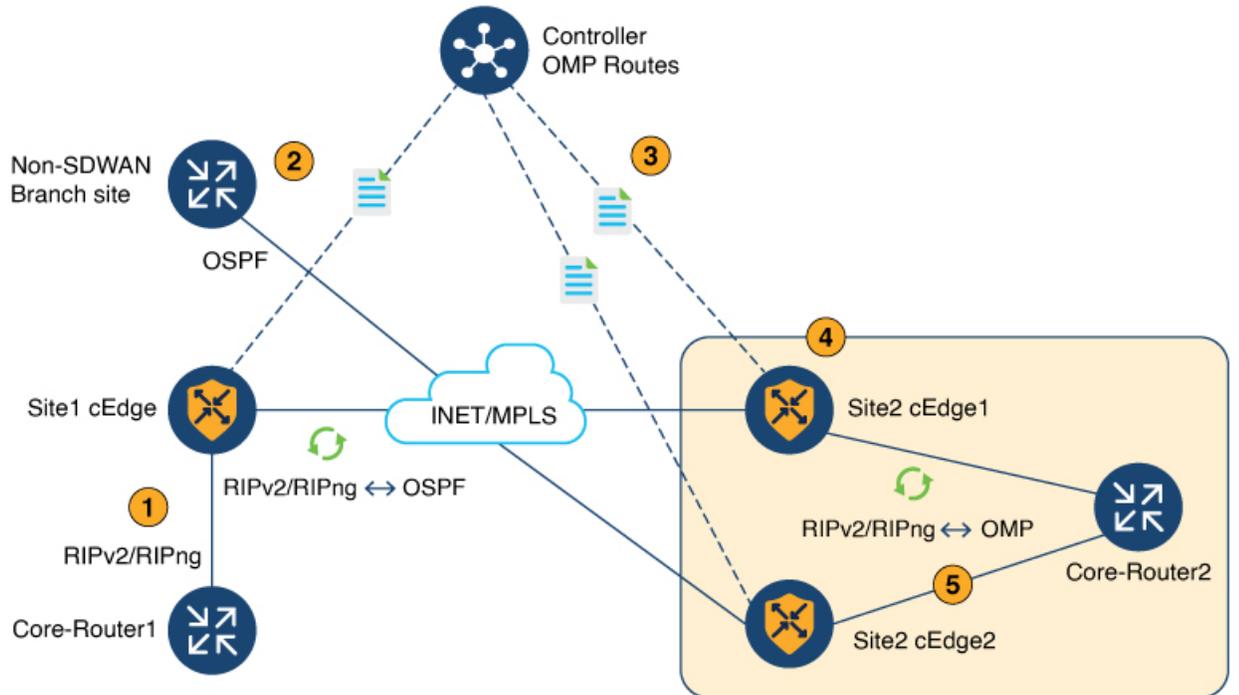
### **RIPng Loop Avoidance**

RIPng in large networks is prone to routing loops, which can result in traffic taking longer paths. To avoid route looping, RIP and RIPng routes are identified using the well-known OMP RIP tag.

RIPv2 and RIPng OMP Route Tagging Process

The following process illustrates how OMP route tagging helps avoid routing loops in RIPv2 and RIPng:

Figure 1: RIPv2 and RIPv2 Topology



1. Core-Router1 advertises RIPv2 and RIPv2 routes to Site1.  
RIPv2 and RIPv2 routes have a default administrative distance of 120, while OMP routes have a default administrative distance of 251.
2. The RIPv2 and RIPv2 route is redistributed and advertised in OMP.
3. The Cisco Catalyst SD-WAN Controller advertises an OMP route to the other branch.
4. Site-2 Edge1 router adds an OMP route tag of a unique value (e.g., 44270) and redistributes the OMP-learned route into RIPv2 and RIPv2.
5. When the Site-2 Edge2 router receives this route with the tag 44270, it will not install this route because it is already learning a route through OMP, which has an administrative distance (AD) of 251 (preferred over 120).

If the OMP route is withdrawn, the Site-2 Edge2 router installs the route learned through the RIPv2 and RIPv2 protocol via a service-side VPN with the tag 44270, into the routing table with an administrative distance of 252 (one value higher than that of OMP).

Additionally, a Cisco Catalyst SD-WAN tagged route will not be readvertised in OMP when the RIPv2 and RIPv2 route is redistributed to OMP.

See [Configure RIPv2 using the CLI, on page 10](#) for more details on RIPv2 configurations using the CLI.

## Prerequisites for using RIP

For proper operation, Version 2 (RIPv2) must be configured to send and receive only RIPv2 packets. By default, RIP Version 1 (RIPv1) and (RIPv2) packets are received, but only RIPv1 packets are sent.

## Restrictions for using RIP

When implementing RIP or RIPng (IPv6), be aware of the following limitations and restrictions:

### RIPv2 (IPv4) restrictions

- 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, while an unreachable network has a metric of 16.
- This limited metric range makes RIP unsuitable for large networks.

### RIPng (IPv6) restrictions

- Only the **sdwan** keyword can be used to configure the IPv6 RIP routing process name (ripng-instance) in the configuration commands.
- VRF-aware support in IPv6 RIP allows only one RIP instance at a given time; more than one RIP instance is not allowed.
- You can configure RIPng on only GigabitEthernet, TenGigabitEthernet, and VLAN interfaces.

## Configure RIPv2 using the CLI

Use this task to configure RIPv2 on Cisco IOS XE Catalyst SD-WAN devices via the Command Line Interface (CLI).

This section provides detailed instructions for configuring RIPv2, including process setup, VRF-aware support, route summarization, and various other RIP parameters. You can apply the configuration using CLI device templates or CLI Add-on feature templates.

### Before you begin

You must complete the initial VRF routing table and address family submode configurations on the device. You can run these commands in any order.

Follow these steps to configure RIPv2 using the CLI:

### Procedure

---

#### Step 1

Configure the RIP routing process.

Enable a RIP routing process and enter router configuration mode.

```
Device# config-transaction
Device(config)# router rip
Device(config-router)#
```

#### Step 2

Configure the RIP VRF-aware support.

Enter VRF address family configuration mode and enable IPv4 address prefixes.

```
Device(config)# router rip
Device(config-router)# address-family ipv4 vrf vrf-name
```

**Step 3** Specify the RIP version.

Specify RIP version as 2 to enable the device to send only RIP version 2 (RIPv2) packets:

```
Device(config)# router rip
Device(config-router)# version {1|2}
```

**Step 4** Configure RIP routes summarization.

Disable or restore the default behavior of automatic summarization of subnet routes into network-level routes used in router configuration mode:

```
Device(config)# router rip
Device(config-router)# auto-summary
```

**Step 5** Validate the source IP address.

Enable a router to perform validation checks on the source IP address of incoming RIP updates:

```
Device(config)# router rip
Device(config-router)# network ip-address
Device(config-router)# validate-update-source
```

**Step 6** Configure interpacket delay.

Configure interpacket delay for outbound RIP updates, in milliseconds:

```
Device(config)# router rip
Device(config-router)# output-delay delay-value
```

**Step 7** Redistribute the routes into the RIP routing process.

Redistribute the specified routes into the IPv4 RIP routing process. Only configure protocol redistribution after you configure the source router protocols. The protocol argument can be one of these keywords.

- **bgp**
- **connected**
- **isis**
- **eigrp**
- **omp**
- **ospf**
- **ospfv3**
- **static**
- **static**

In Cisco IOS XE Catalyst SD-WAN Release 17.7.1aa, RIP Version 2 configurations in Cisco IOS XE Catalyst SD-WAN devices support OMP as a redistributed protocol.

```
Device(config)# router rip
Device(config-router)# redistribute protocol [metric metric-value] [route-map map-name]
```

**Step 8** Filter the RIP-routing updates.

Apply a prefix list to the RIP-routing updates that are received in or sent over an interface:

```
Device(config)# router rip
Device(config-router)# distribute-list prefix-list listname {in | out} [interface-type
interface-number]
```

**Step 9** Configure the RIP parameters.

The network command is required to enable interfaces for RIP(v2), and to associate a network with a RIP routing process. There's no limit on the number of network commands that you can use on the router. Use classful (Class A, Class B, Class C) IP network ID addressing for network configurations.

```
Device(config)# router rip
Device(config-router)# network ip-address
```

a) Define a neighboring device with which to exchange routing information.

```
Device(config)# router rip
Device(config-router)# network ip-address
Device(config-router)# neighbor ip-address bfd
```

b) Apply an offset list to routing metrics:

```
Device(config)# router rip
Device(config-router)# offset-list acl-number in offset[ interface-type |interface-name]
```

c) Adjust routing protocol timers:

```
Device(config)# router rip
Device(config-router)# timers basic update invalid holddown flush
```

**Step 10** Customize a RIP.

Define the maximum number of equal-cost routes that an IPv4 RIP can support:

```
Device(config)# router rip
Device(config-router)# maximum-paths number-paths
```

**Step 11** Configure a route tag.

By default, automatic RIPv2 route tag is enabled for redistributed OMP routes. When a router is installed by another Cisco IOS XE Catalyst SD-WAN device, the admin distance is set to 252 so that OMP routes are preferred over redistributed OMP routes:

```
Device(config)# router rip
Device(config-router)# omp-route-tag
```

**Step 12** Configure the traffic.

Configure traffic to use minimum-cost paths, and load splitting on multiinterfaces with equal-cost paths:

```
Device(config)# router rip
Device(config-router)# traffic-share min across-interfaces
```

---

RIPv2 is configured on the Cisco IOS XE Catalyst SD-WAN device.

**What to do next**

Verify the RIPv2 configurations using the CLI. For more details, see the [Verification commands for RIPv2 configuration, on page 8](#).

The following is a complete example of RIPv2 configuration for Cisco IOS XE Catalyst SD-WAN devices using the CLI:

```
config-transaction
!
    vrf definition 172
    address-family ipv4
    exit-address-family
!
    router rip
    address-family ipv4 vrf 172
    distance 70
    omp-route-tag /* Default is enabled */
    default-information originate route-map RIP-MED
    version 2
    network 10.0.0.20 /* Only classful A, B, or C network. */
    distribute-list prefix v4KANYU-RIP in TenGigabitEthernet0/1/3.791
    redistribute rip v6kanyu metric 1 metric-type 1 route-map v6RED-RIP-OSPF1
    distribute-list prefix v4KANYU-RIP in TenGigabitEthernet0/1/3.792
    no auto-summary
!
```

## Verification commands for RIPv2 configuration

This section details the essential verification commands used to confirm the RIPv2 configurations and operational status on Cisco IOS XE Catalyst SD-WAN devices. These commands are crucial for inspecting configured settings, routing table entries, RIP database details, and neighbor information to ensure proper RIP functionality within the network.

### View RIPv2 routing configurations in the running configuration

```
Device# show sdwan running | sec rip
router rip
    version 2
    redistribute connected
    output-delay 20
    input-queue 20
!
address-family ipv4 vrf 200
    redistribute connected
    redistribute omp metric 2
    network 56.0.0.0
    no auto-summary
    version 2
    exit-address-family
```

### Display RIPv2 routes in the default routing table

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

```

    & - replicated local route overrides by connected

Gateway of last resort is 10.0.5.13 to network 10.10.10.10

R    10.11.0.0/16 [120/1] via 172.16.1.2, 00:00:02, GigabitEthernet1

```

### Display RIP routes under a specific VRF table

```

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

Gateway of last resort is not set
10.0.0.14/32 is subnetted, 1 subnets
R 10.14.14.14 [120/1] via 10.20.25.18, 00:00:18, GigabitEthernet5

```

### Display the contents of the RIP private database

```

Device# show ip rip database
10.11.0.0/16    auto-summary
10.11.0.0/16
[1] via 172.16.1.2, 00:00:00, GigabitEthernet1

```

### Display RIP Bidirectional Forwarding Detection (BFD) neighbors

```

Device# show ip rip neighbors
BFD sessions created for the RIP neighbors
Neighbor      Interface      SessionHandle
10.10.10.2    GigabitEthernet1  1

```

### Display RIP protocol configurations

```

Device# show ip protocols | sec rip
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 19 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Neighbor(s):
    10.1.1.2
  Default version control: send version 2, receive version 2
  Interface          Send  Recv  Triggered RIP  Key-chain
  GigabitEthernet1    2     2     No              none
  Loopback10          2     2     No              none
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.11.0.1
  Routing Information Sources:
    Gateway          Distance      Last Update

```

```
10.1.1.2          120          00:00:15
Distance: (default is 120)
```

## Configure RIPng using the CLI

Use this task to configure Routing Information Protocol next generation (RIPng) on Cisco IOS XE Catalyst SD-WAN devices via the Command Line Interface (CLI).

This section explains how to configure RIPng. It covers VRF-aware support, routing process setup, route tagging, and interface-specific parameters. You can use CLI device templates and CLI Add-on feature templates to set up RIPng.

### Before you begin

Initial VRF routing table and address family submode configurations are required to verify RIPng configurations using the **show ipv6 route vrf** command.

Use these steps to configure RIPng using the CLI::

### Procedure

**Step 1** Configure IPv6 RIPng VRF-aware support.

- a) Enable VRF-aware support for IPv6 RIPng routing.

It is mandatory for the RIPng to be configured within the service VPN.

```
Device(config)# ipv6 rip vrf-mode enable
```

- b) Enable the forwarding of IPv6 unicast datagrams:

```
Device(config)# ipv6 unicast-routing
```

**Step 2** Configure the IPv6 RIPng routing process and enable router configuration mode for the IPv6 RIPng routing process:

```
Device(config)# ipv6 router rip ripng-instance
Device(config-rtr)#
```

For **ripng-instance**, use **sdwan**.

**Step 3** Enter VRF address family configuration mode and enable IPv6 address prefixes:

```
Device(config)# ipv6 router rip ripng-instance
Device(config-rtr)# address-family ipv6 vrf vrf-name
Device(config-ipv6-router-af)#
```

**Step 4** Define an administrative distance for routes that are inserted into a routing table:

```
Device(config)# ipv6 router rip ripng-instance
Device(config-rtr)# address-family ipv6 vrf vrf-name
Device(config-ipv6-router-af)# distance distance
```

**Step 5** Configure a route tag.

By default, automatic RIPng route tagging is enabled for redistributed OMP routes. When a Cisco IOS XE Catalyst SD-WAN device learns a RIPv2 or RIPng route with a unique SD-WAN tag (44270), the router installs the route with an administrative distance of 252. This value is higher than the OMP distance (251), so the OMP routes are preferred over redistributed OMP routes.

```
Device(config)# ipv6 router rip ripng-instance
Device(config-rtr)# omp-route-tag
```

**Step 6** Create an entry in the IPv6 prefix list:

```
Device(config)# ipv6 prefix-list list-name [seq seq-number] permit IPv6 prefix (IP/length)
```

**Step 7** Apply a prefix list to IPv6 RIPng routing updates that are received or sent on an interface:

```
Device(config)# ipv6 router rip ripng-instance
Device(config-rtr)# distribute-list prefix-list prefix-list-name {in | out} [interface-type |
interface-number]
```

**Step 8** Redistribute the specified routes into the IPv6 RIPng routing process. The rip keyword and ripng-instance specify an IPv6 RIPng routing process.

```
Device(config)# ipv6 router rip ripng-instance
Device(config-rtr)# redistribute protocol [metric default-metric] [route-map map-tag]
```

**Step 9** Configure the interface.

## a) Enable the specified IPv6 RIPng routing process on an interface:

For **ripng-instance**, use **sdwan**.

```
Device(config)# interface type number
Device(config-if)# ipv6 rip ripng-instance enable
```

## b) (Optional) The IPv6 default route (::/0) distributes into the specified RIPng routing process updates sent out of the specified interface:

For **ripng-instance**, use **sdwan**.

```
Device(config)# interface type number
Device(config-if)# ipv6 rip ripng-instance default-information {only | originate} [metric
metric-value]
```

## c) Set the IPv6 RIPng metric-offset for an interface.

For **ripng-instance**, use **sdwan**.

```
Device(config)# interface type number
Device(config-if)# ipv6 rip ripng-instance metric-offset metric-value
```

## d) Configure the IPv6 RIPng to advertise summarized IPv6 addresses on an interface and to specify the IPv6 prefix that identifies the routes to be summarized.

For **ripng-instance**, use **sdwan**.

```
Device(config)# interface type number
Device(config-if)# ipv6 address {ipv6-prefix/prefix-length | prefix-name | sub-bits/prefix-length}

Device(config-if)# ipv6 rip ripng-instance summary-address {ipv6-prefix/prefix-length}
```

---

RIPng is configured on the Cisco IOS XE Catalyst SD-WAN device

**What to do next**

Verify the RIPng configurations using the CLI. For more details, see the [Verification commands for RIPng configuration, on page 12](#).

The following example shows a complete RIPng configuration for Cisco IOS XE Catalyst SD-WAN devices using the CLI:

```
config-transaction
!
  vrf definition 1
    address-family ipv6
    exit-address-family
!
  ipv6 rip vrf-mode enable
  ipv6 unicast-routing
!
  ipv6 prefix-list cisco seq 10 permit 2000:1::/64
!
  ipv6 router rip sdwan
    address-family ipv6 vrf 1
      distance 130
      omp-route-tag
      distribute-list prefix-list cisco in GigabitEthernet0/0/0
      redistribute omp metric 10
      exit-address-family
!
  interface GigabitEthernet0/0/0
    ipv6 address 2001:DB8::/64
    ipv6 rip sdwan enable
    ipv6 rip sdwan default-information originate
    ipv6 rip sdwan metric-offset 10
    ipv6 rip sdwan summary-address 2001:90::1/32
!
```

## Verification commands for RIPng configuration

This section details the essential verification commands used to confirm the Routing Information Protocol next generation (RIPng) configurations and operational status on Cisco IOS XE Catalyst SD-WAN devices. These commands are crucial for inspecting configured settings and routing table entries to ensure proper RIPng functionality within the network.

### Display RIPng routes in the VRF routing table

The following is a sample output from the show ipv6 route vrf command displaying the router RIPng configurations:

```
Device# show ipv6 route vrf 1
IPv6 Routing Table - 1 - 11 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, RL - RPL, O - OSPF Intra, OI - OSPF Inter
       OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1
       ON2 - OSPF NSSA ext 2, la - LISP alt, lr - LISP site-registrations
       ld - LISP dyn-eid, la - LISP away, le - LISP extranet-policy
       lp - LISP publications, ls - LISP destinations-summary, a - Application
       m - OMP
R 1100::/64 [120/2]
  via FE80::20C:29FF:FE2E:13FF, GigabitEthernet2
R 2000::/64 [120/2]
  via FE80::20C:29FF:FE51:762F, GigabitEthernet2
R 2001:10::/64 [120/2]
  via FE80::20C:29FF:FE82:D659, GigabitEthernet2
```

```
R 2500::/64 [252/11], tag 44270
   via FE80::20C:29FF:FEE1:5237, GigabitEthernet2
C 2750::/64 [0/0]
   via GigabitEthernet2, directly connected
L 2750::1/128 [0/0]
   via GigabitEthernet2, receive
R 2777::/64 [252/11], tag 44270
   via FE80::20C:29FF:FEE1:5237, GigabitEthernet2
m 2900::/64 [251/0]
   via 192.168.1.5%default
R 3000::/64 [120/2]
   via FE80::20C:29FF:FE2E:13FF, GigabitEthernet2
R 3400::/64 [252/11], tag 44270
   via FE80::20C:29FF:FE51:762F, GigabitEthernet2
L FF00::/8 [0/0]
   via Null0, receive
```

