



Hewlett Packard
Enterprise

HPE FlexNetwork 5130 EI Switch Series

Layer 3—IP Routing Configuration Guide

Part number: 5998-5483s
Software version: Release 3111P02 and later
Document version: 6W101-20161010

© Copyright 2015, 2016 Hewlett Packard Enterprise Development LP

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

Microsoft® and Windows® are trademarks of the Microsoft group of companies.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java and Oracle are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Contents

Configuring basic IP routing	1
Routing table	1
Dynamic routing protocols	2
Route preference	2
Route backup	2
Route recursion	2
Route redistribution	3
Configuring the maximum lifetime for routes and labels in the RIB	3
Configuring the maximum lifetime for routes in the FIB	3
Displaying and maintaining a routing table	4
Configuring static routing	6
Configuring a static route	6
Configuring BFD for static routes	6
Bidirectional control mode	7
Single-hop echo mode	7
Displaying and maintaining static routes	8
Static route configuration examples	8
Basic static route configuration example	8
BFD for static routes configuration example (direct next hop)	10
BFD for static routes configuration example (indirect next hop)	12
Configuring a default route	16
Configuring RIP	17
Overview	17
RIP route entries	17
Routing loop prevention	17
RIP operation	17
RIP versions	18
Protocols and standards	18
RIP configuration task list	18
Configuring basic RIP	19
Enabling RIP	19
Controlling RIP reception and advertisement on interfaces	20
Configuring a RIP version	20
Configuring RIP route control	21
Configuring an additional routing metric	21
Configuring RIPv2 route summarization	21
Disabling host route reception	22
Advertising a default route	23
Configuring received/redistributed route filtering	23
Configuring a preference for RIP	24
Configuring RIP route redistribution	24
Tuning and optimizing RIP networks	24
Configuration prerequisites	24
Configuring RIP timers	24
Configuring split horizon and poison reverse	25
Enabling zero field check on incoming RIPv1 messages	26
Enabling source IP address check on incoming RIP updates	26
Configuring RIPv2 message authentication	26
Specifying a RIP neighbor	27
Configuring RIP network management	27
Configuring the RIP packet sending rate	28
Setting the maximum length of RIP packets	28
Configuring RIP GR	28
Configuring BFD for RIP	29

Configuring single-hop echo detection (for a directly connected RIP neighbor)	29
Configuring single-hop echo detection (for a specific destination)	30
Configuring bidirectional control detection	30
Displaying and maintaining RIP	30
RIP configuration examples	31
Basic RIP configuration example	31
RIP route redistribution configuration example	34
RIP interface additional metric configuration example	36
BFD for RIP configuration example (single-hop echo detection for a directly connected neighbor)	37
BFD for RIP configuration example (single hop echo detection for a specific destination)	40
BFD for RIP configuration example (bidirectional detection in BFD control packet mode)	43
Configuring PBR	47
Overview	47
Policy	47
PBR and Track	48
PBR configuration task list	48
Configuring a policy	48
Creating a node	48
Configuring match criteria for a node	48
Configuring actions for a node	49
Configuring PBR	49
Configuring local PBR	49
Configuring interface PBR	50
Displaying and maintaining PBR	50
PBR configuration examples	50
Packet type-based local PBR configuration example	50
Packet type-based interface PBR configuration example	52
Configuring IPv6 static routing	55
Configuring an IPv6 static route	55
Configuring BFD for IPv6 static routes	55
Bidirectional control mode	55
Single-hop echo mode	56
Displaying and maintaining IPv6 static routes	57
IPv6 static routing configuration examples	57
Basic IPv6 static route configuration example	57
BFD for IPv6 static routes configuration example (direct next hop)	59
BFD for IPv6 static routes configuration example (indirect next hop)	61
Configuring an IPv6 default route	65
Configuring RIPng	66
Overview	66
RIPng route entries	66
RIPng packets	66
Protocols and standards	67
RIPng configuration task list	67
Configuring basic RIPng	67
Configuring RIPng route control	68
Configuring an additional routing metric	68
Configuring RIPng route summarization	68
Advertising a default route	69
Configuring received/redistributed route filtering	69
Configuring a preference for RIPng	69
Configuring RIPng route redistribution	70
Tuning and optimizing the RIPng network	70
Configuring RIPng timers	70
Configuring split horizon and poison reverse	70
Configuring zero field check on RIPng packets	71
Configuring RIPng GR	71
Applying an IPsec profile	72

Displaying and maintaining RIPng	72
RIPng configuration examples	73
Basic RIPng configuration example	73
RIPng route redistribution configuration example	75
RIPng IPsec profile configuration example	78
Configuring IPv6 PBR	81
Overview	81
Policy	81
PBR and Track	82
IPv6 PBR configuration task list	82
Configuring an IPv6 policy	82
Creating an IPv6 node	82
Configuring match criteria for an IPv6 node	82
Configuring actions for an IPv6 node	83
Configuring IPv6 PBR	83
Configuring IPv6 local PBR	83
Configuring IPv6 interface PBR	84
Displaying and maintaining IPv6 PBR	84
IPv6 PBR configuration examples	84
Packet type-based IPv6 local PBR configuration example	84
Packet type-based IPv6 interface PBR configuration example	86
Configuring routing policies	89
Overview	89
Filters	89
Routing policy	89
Configuring filters	90
Configuration prerequisites	90
Configuring an IP prefix list	90
Configuring a routing policy	91
Configuration prerequisites	91
Creating a routing policy	91
Configuring if-match clauses	91
Configuring apply clauses	92
Configuring the continue clause	93
Displaying and maintaining the routing policy	93
Routing policy configuration example for IPv6 route redistribution	93
Network requirements	93
Configuration procedure	94
Verifying the configuration	95
Document conventions and icons	96
Conventions	96
Network topology icons	97
Support and other resources	98
Accessing Hewlett Packard Enterprise Support	98
Accessing updates	98
Websites	99
Customer self repair	99
Remote support	99
Documentation feedback	99
Index	101

Configuring basic IP routing

The term "interface" in this chapter collectively refers to Layer 3 interfaces, including VLAN interfaces.

IP routing directs IP packet forwarding on routers based on a routing table. This chapter focuses on unicast routing protocols. For more information about multicast routing protocols, see *IP Multicast Configuration Guide*.

Routing table

A RIB contains the global routing information and related information, including route recursion, route redistribution, and route extension information. The router selects optimal routes from the routing table and puts them into the FIB table. It uses the FIB table to forward packets. For more information about the FIB table, see *Layer 3—IP Services Configuration Guide*.

Table 1 categorizes routes by different criteria.

Table 1 Route categories

Criterion	Categories
Destination	<ul style="list-style-type: none">• Network route—The destination is a network. The subnet mask is less than 32 bits.• Host route—The destination is a host. The subnet mask is 32 bits.
Whether the destination is directly connected	<ul style="list-style-type: none">• Direct route—The destination is directly connected.• Indirect route—The destination is indirectly connected.
Origin	<ul style="list-style-type: none">• Direct route—A direct route is discovered by the data link protocol on an interface, and is also called an interface route.• Static route—A static route is manually configured by an administrator.• Dynamic route—A dynamic route is dynamically discovered by a routing protocol.

To view brief information about a routing table, use the **display ip routing-table** command:

```
<Sysname> display ip routing-table
```

```
Destinations : 19          Routes : 19

Destination/Mask    Proto  Pre  Cost           NextHop           Interface
0.0.0.0/32          Direct  0    0              127.0.0.1         InLoop0
1.1.1.0/24          Direct  0    0              1.1.1.1           Vlan1
1.1.1.0/32          Direct  0    0              1.1.1.1           Vlan1
1.1.1.1/32          Direct  0    0              127.0.0.1         InLoop0
1.1.1.255/32        Direct  0    0              1.1.1.1           Vlan1
2.2.2.0/24          Static  60   0              12.2.2.2          Vlan2
...
```

A route entry includes the following key items:

- **Destination**—IP address of the destination host or network.
- **Mask**—Mask length of the IP address.
- **Pre**—Preference of the route. Among routes to the same destination, the route with the highest preference is optimal.

- **Cost**—If multiple routes to a destination have the same preference, the one with the smallest cost is the optimal route.
- **NextHop**—Next hop.
- **Interface**—Output interface.

Dynamic routing protocols

Static routes work well in small, stable networks. They are easy to configure and require fewer system resources. However, in networks where topology changes occur frequently, a typical practice is to configure a dynamic routing protocol. Compared with static routing, a dynamic routing protocol is complicated to configure, requires more router resources, and consumes more network resources.

Dynamic routing protocols dynamically collect and report reachability information to adapt to topology changes. They are suitable for large networks.

An AS refers to a group of routers that use the same routing policy and work under the same administration.

Route preference

Routing protocols, including static and direct routing, each by default have a preference. If they find multiple routes to the same destination, the router selects the route with the highest preference as the optimal route.

The preference of a direct route is always 0 and cannot be changed. You can configure a preference for each static route and each dynamic routing protocol. The following table lists the route types and default preferences. The smaller the value, the higher the preference.

Table 2 Route types and default route preferences

Route type	Preference
Direct route	0
Multicast static route	1
Unicast static route	60
RIP	100
Unknown (route from an untrusted source)	256

Route backup

Route backup can improve network availability. Among multiple routes to the same destination, the route with the highest priority is the primary route and others are secondary routes.

The router forwards matching packets through the primary route. When the primary route fails, the route with the highest preference among the secondary routes is selected to forward packets. When the primary route recovers, the router uses it to forward packets.

Route recursion

To use a route that has an indirectly connected next hop, a router must perform route recursion to find the output interface to reach the next hop.

The RIB records and saves route recursion information, including brief information about related routes, recursive paths, and recursion depth.

Route redistribution

Route redistribution enables routing protocols to learn routing information from each other. A dynamic routing protocol can redistribute routes from other routing protocols, including direct and static routing. For more information, see the respective chapters on those routing protocols in this configuration guide.

The RIB records redistribution relationships of routing protocols.

Configuring the maximum lifetime for routes and labels in the RIB

Perform this task to prevent routes of a certain protocol from being aged out due to slow protocol convergence resulting from a large number of route entries or long GR period.

The configuration takes effect at the next protocol or RIB process switchover.

To configure the maximum lifetime for routes and labels in the RIB (IPv4):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIB view.	rib	N/A
3. Create a RIB IPv4 address family and enter RIB IPv4 address family view.	address-family ipv4	By default, no RIB IPv4 address family is created.
4. Configure the maximum lifetime for IPv4 routes and labels in the RIB.	protocol protocol lifetime seconds	By default, the maximum lifetime for routes and labels in the RIB is 480 seconds.

To configure the maximum route lifetime for routes and labels in the RIB (IPv6):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIB view.	rib	N/A
3. Create a RIB IPv6 address family and enter RIB IPv6 address family view.	address-family ipv6	By default, no RIB IPv6 address family is created.
4. Configure the maximum lifetime for IPv6 routes and labels in the RIB.	protocol protocol lifetime seconds	By default, the maximum lifetime for routes and labels in the RIB is 480 seconds.

Configuring the maximum lifetime for routes in the FIB

When GR or NSR is disabled, FIB entries must be retained for some time after a protocol process switchover or RIB process switchover. When GR or NSR is enabled, FIB entries must be removed

immediately after a protocol or RIB process switchover to avoid routing issues. Perform this task to meet such requirements.

To configure the maximum lifetime for routes in the FIB (IPv4):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIB view.	rib	N/A
3. Create a RIB IPv4 address family and enter its view.	address-family ipv4	By default, no RIB IPv4 address family is created.
4. Configure the maximum lifetime for IPv4 routes in the FIB.	fib lifetime seconds	By default, the maximum lifetime for routes in the FIB is 600 seconds.

To configure the maximum lifetime for routes in the FIB (IPv6):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIB view.	rib	N/A
3. Create a RIB IPv6 address family and enter its view.	address-family ipv6	By default, no RIB IPv6 address family is created.
4. Configure the maximum lifetime for IPv6 routes in the FIB.	fib lifetime seconds	By default, the maximum lifetime for routes in the FIB is 600 seconds.

Displaying and maintaining a routing table

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display routing table information.	display ip routing-table [verbose]
Display information about routes permitted by an IPv4 basic ACL.	display ip routing-table acl acl-number [verbose]
Display information about routes to a specific destination address.	display ip routing-table ip-address [mask mask-length] [longer-match] [verbose]
Display information about routes to a range of destination addresses.	display ip routing-table ip-address1 to ip-address2 [verbose]
Display information about routes permitted by an IP prefix list.	display ip routing-table prefix-list prefix-list-name [verbose]
Display information about routes installed by a protocol.	display ip routing-table protocol protocol [inactive verbose]
Display IPv4 route statistics.	display ip routing-table statistics
Display brief IPv4 routing table information.	display ip routing-table summary
Display route attribute information in the RIB.	display rib attribute [attribute-id]
Display RIB GR state information.	display rib graceful-restart

Task	Command
Display next hop information in the RIB.	display rib nib [self-originated] [nib-id] [verbose] display rib nib protocol <i>protocol-name</i> [verbose]
Display next hop information for direct routes.	display route-direct nib [nib-id] [verbose]
Clear IPv4 route statistics.	reset ip routing-table statistics protocol { <i>protocol</i> all }
Display IPv6 routing table information.	display ipv6 routing-table [verbose]
Display information about routes to an IPv6 destination address.	display ipv6 routing-table <i>ipv6-address</i> [<i>prefix-length</i>] [longer-match] [verbose]
Display information about routes permitted by an IPv6 basic ACL.	display ipv6 routing-table acl <i>acl6-number</i> [verbose]
Display information about routes to a range of IPv6 destination addresses.	display ipv6 routing-table <i>ipv6-address1</i> to <i>ipv6-address2</i> [verbose]
Display information about routes permitted by an IPv6 prefix list.	display ipv6 routing-table prefix-list <i>prefix-list-name</i> [verbose]
Display information about routes installed by an IPv6 protocol.	display ipv6 routing-table protocol <i>protocol</i> [inactive verbose]
Display IPv6 route statistics.	display ipv6 routing-table statistics
Display brief IPv6 routing table information.	display ipv6 routing-table summary
Display route attribute information in the IPv6 RIB.	display ipv6 rib attribute [<i>attribute-id</i>]
Display IPv6 RIB GR state information.	display ipv6 rib graceful-restart
Display next hop information in the IPv6 RIB.	display ipv6 rib nib [self-originated] [nib-id] [verbose] display ipv6 rib nib protocol <i>protocol-name</i> [verbose]
Display next hop information for IPv6 direct routes.	display ipv6 route-direct nib [nib-id] [verbose]
Clear IPv6 route statistics.	reset ipv6 routing-table statistics protocol { <i>protocol</i> all }

Configuring static routing

Static routes are manually configured. If a network's topology is simple, you only need to configure static routes for the network to work correctly.

Static routes cannot adapt to network topology changes. If a fault or a topological change occurs in the network, the network administrator must modify the static routes manually.

Configuring a static route

Before you configure a static route, complete the following tasks:

- Configure the physical parameters for related interfaces.
- Configure the link-layer attributes for related interfaces.
- Configure the IP addresses for related interfaces.

You can associate Track with a static route to monitor the reachability of the next hops. For more information about Track, see *High Availability Configuration Guide*.

To configure a static route:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure a static route.	ip route-static <i>dest-address</i> { <i>mask-length</i> <i>mask</i> } { <i>interface-type</i> <i>interface-number</i> [<i>next-hop-address</i>] <i>next-hop-address</i> [track <i>track-entry-number</i>] } [permanent] [preference <i>preference-value</i>] [tag <i>tag-value</i>] [description <i>description-text</i>]	By default, no static route is configured.
3. (Optional.) Configure the default preference for static routes.	ip route-static default-preference <i>default-preference-value</i>	The default setting is 60.
4. (Optional.) Delete all static routes, including the default route.	delete static-routes all	To delete one static route, use the undo ip route-static command.

Configuring BFD for static routes

ⓘ IMPORTANT:

Enabling BFD for a flapping route could worsen the situation.

BFD provides a general-purpose, standard, medium-, and protocol-independent fast failure detection mechanism. It can uniformly and quickly detect the failures of the bidirectional forwarding paths between two routers for protocols, such as routing protocols.

For more information about BFD, see *High Availability Configuration Guide*.

Bidirectional control mode

To use BFD bidirectional control detection between two devices, enable BFD control mode for each device's static route destined to the peer.

To configure a static route and enable BFD control mode, use one of the following methods:

- Specify an output interface and a direct next hop.
- Specify an indirect next hop and a specific BFD packet source address for the static route.

To configure BFD control mode for a static route (direct next hop):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure BFD control mode for a static route.	ip route-static <i>dest-address</i> { <i>mask-length</i> <i>mask</i> } <i>interface-type interface-number next-hop-address</i> bfd control-packet [preference <i>preference-value</i>] [tag <i>tag-value</i>] [description <i>description-text</i>]	By default, BFD control mode for a static route is not configured.

To configure BFD control mode for a static route (indirect next hop):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure BFD control mode for a static route.	ip route-static <i>dest-address</i> { <i>mask-length</i> <i>mask</i> } { <i>next-hop-address</i> bfd control-packet bfd-source <i>ip-address</i> } [preference <i>preference-value</i>] [tag <i>tag-value</i>] [description <i>description-text</i>]	By default, BFD control mode for a static route is not configured.

Single-hop echo mode

With BFD echo mode enabled for a static route, the output interface sends BFD echo packets to the destination device, which loops the packets back to test the link reachability.

ⓘ IMPORTANT:

Do not use BFD for a static route with the output interface in spoofing state.

To configure BFD echo mode for a static route:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure the source address of echo packets.	bfd echo-source-ip <i>ip-address</i>	By default, the source address of echo packets is not configured.

Step	Command	Remarks
		For more information about this command, see <i>High Availability Command Reference</i> .
3. Configure BFD echo mode for a static route.	ip route-static dest-address { mask-length mask } interface-type interface-number next-hop-address bfd echo-packet [preference preference-value] [tag tag-value] [description description-text]	By default, BFD echo mode for a static route is not configured.

Displaying and maintaining static routes

Execute **display** commands in any view.

Task	Command
Display static route information.	display ip routing-table protocol static [inactive verbose]
Display static route next hop information.	display route-static nib [nib-id] [verbose]
Display static routing table information.	display route-static routing-table [ip-address { mask-length mask }]

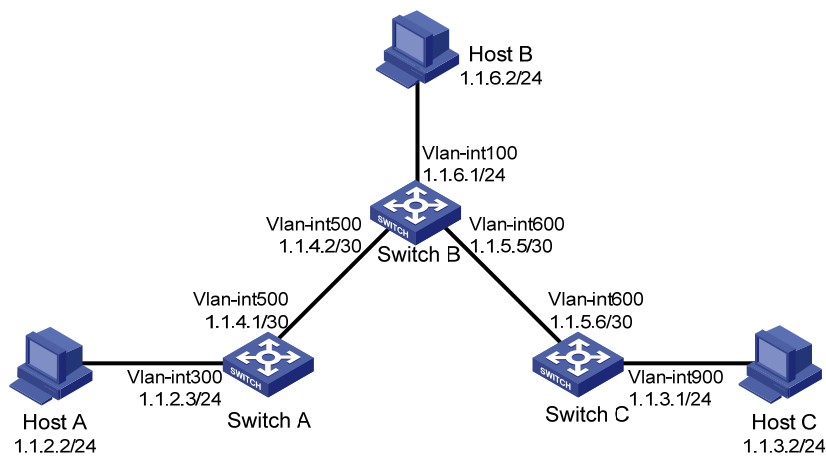
Static route configuration examples

Basic static route configuration example

Network requirements

As shown in [Figure 1](#), configure static routes on the switches for interconnections between any two hosts.

Figure 1 Network diagram



Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)
2. Configure static routes:
 - # Configure a default route on Switch A.
<SwitchA> system-view
[SwitchA] ip route-static 0.0.0.0 0.0.0.0 1.1.4.2
 - # Configure two static routes on Switch B.
<SwitchB> system-view
[SwitchB] ip route-static 1.1.2.0 255.255.255.0 1.1.4.1
[SwitchB] ip route-static 1.1.3.0 255.255.255.0 1.1.5.6
 - # Configure a default route on Switch C.
<SwitchC> system-view
[SwitchC] ip route-static 0.0.0.0 0.0.0.0 1.1.5.5
3. Configure the default gateways of Host A, Host B, and Host C as 1.1.2.3, 1.1.6.1, and 1.1.3.1. (Details not shown.)

Verifying the configuration

Display static routes on Switch A.

```
[SwitchA] display ip routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/0	Static	60	0	1.1.4.2	Vlan500

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

Display static routes on Switch B.

```
[SwitchB] display ip routing-table protocol static
```

```
Summary Count : 2
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 2
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
1.1.2.0/24	Static	60	0	1.1.4.1	Vlan500

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

Use the **ping** command on Host B to test the reachability of Host A (Windows XP runs on the two hosts).

```
C:\Documents and Settings\Administrator>ping 1.1.2.2
```

```
Pinging 1.1.2.2 with 32 bytes of data:
```

```

Reply from 1.1.2.2: bytes=32 time=1ms TTL=126
Reply from 1.1.2.2: bytes=32 time=1ms TTL=126
Reply from 1.1.2.2: bytes=32 time=1ms TTL=126
Reply from 1.1.2.2: bytes=32 time=1ms TTL=126

```

Ping statistics for 1.1.2.2:

```

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 1ms, Average = 1ms

```

Use the **tracert** command on Host B to test the reachability of Host A.

```
C:\Documents and Settings\Administrator>tracert 1.1.2.2
```

Tracing route to 1.1.2.2 over a maximum of 30 hops

```

  0  <1 ms    <1 ms    <1 ms    1.1.6.1
  1  <1 ms    <1 ms    <1 ms    1.1.4.1
  2   1 ms    <1 ms    <1 ms    1.1.2.2

```

Trace complete.

BFD for static routes configuration example (direct next hop)

Network requirements

As shown in [Figure 2](#):

- Configure a static route to subnet 120.1.1.0/24 on Switch A.
- Configure a static route to subnet 121.1.1.0/24 on Switch B.
- Enable BFD for both routes.
- Configure a static route to subnet 120.1.1.0/24 and a static route to subnet 121.1.1.0/24 on Switch C.

When the link between Switch A and Switch B through the Layer 2 switch fails, BFD can detect the failure immediately. Switch A then communicates with Switch B through Switch C.

Figure 2 Network diagram

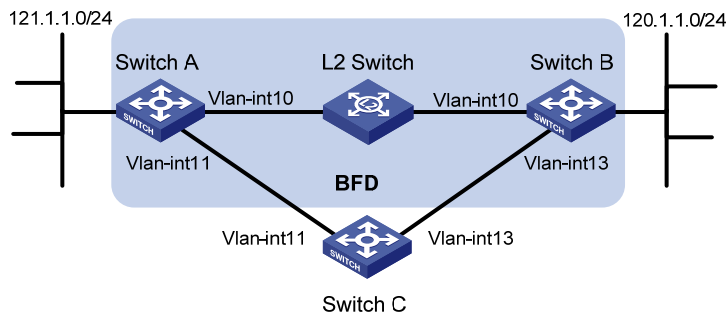


Table 3 Interface and IP address assignment

Device	Interface	IP address
Switch A	VLAN-interface 10	12.1.1.1/24

Device	Interface	IP address
Switch A	VLAN-interface 11	10.1.1.102/24
Switch B	VLAN-interface 10	12.1.1.2/24
Switch B	VLAN-interface 13	13.1.1.1/24
Switch C	VLAN-interface 11	10.1.1.100/24
Switch C	VLAN-interface 13	13.1.1.2/24

Configuration procedure

1. Configure IP addresses for the interfaces. (Details not shown.)
2. Configure static routes and BFD:

Configure static routes on Switch A and enable BFD control mode for the static route that traverses the Layer 2 switch.

```
<SwitchA> system-view
[SwitchA] interface vlan-interface 10
[SwitchA-vlan-interface10] bfd min-transmit-interval 500
[SwitchA-vlan-interface10] bfd min-receive-interval 500
[SwitchA-vlan-interface10] bfd detect-multiplier 9
[SwitchA-vlan-interface10] quit
[SwitchA] ip route-static 120.1.1.0 24 vlan-interface 10 12.1.1.2 bfd control-packet
[SwitchA] ip route-static 120.1.1.0 24 vlan-interface 11 10.1.1.100 preference 65
[SwitchA] quit
```

Configure static routes on Switch B and enable BFD control mode for the static route that traverses the Layer 2 switch.

```
<SwitchB> system-view
[SwitchB] interface vlan-interface 10
[SwitchB-vlan-interface10] bfd min-transmit-interval 500
[SwitchB-vlan-interface10] bfd min-receive-interval 500
[SwitchB-vlan-interface10] bfd detect-multiplier 9
[SwitchB-vlan-interface10] quit
[SwitchB] ip route-static 121.1.1.0 24 vlan-interface 10 12.1.1.1 bfd control-packet
[SwitchB] ip route-static 121.1.1.0 24 vlan-interface 13 13.1.1.2 preference 65
[SwitchB] quit
```

Configure static routes on Switch C.

```
<SwitchC> system-view
[SwitchC] ip route-static 120.1.1.0 24 13.1.1.1
[SwitchC] ip route-static 121.1.1.0 24 10.1.1.102
```

Verifying the configuration

Display BFD sessions on Switch A.

```
<SwitchA> display bfd session
```

```
Total Session Num: 1      Up Session Num: 1      Init Mode: Active
```

```
IPv4 Session Working Under Ctrl Mode:
```

LD/RD	SourceAddr	DestAddr	State	Holdtime	Interface
4/7	12.1.1.1	12.1.1.2	Up	2000ms	Vlan10

The output shows that the BFD session has been created.

Display the static routes on Switch A.

```
<SwitchA> display ip routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
120.1.1.0/24	Static	60	0	12.1.1.2	Vlan10

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

The output shows that Switch A communicates with Switch B through VLAN-interface 10. Then the link over VLAN-interface 10 fails.

Display static routes on Switch A.

```
<SwitchA> display ip routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
120.1.1.0/24	Static	65	0	10.1.1.100	Vlan11

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

The output shows that Switch A communicates with Switch B through VLAN-interface 11.

BFD for static routes configuration example (indirect next hop)

Network requirements

As shown in [Figure 3](#):

- Switch A has a route to interface Loopback 1 (2.2.2.9/32) on Switch B, with the output interface VLAN-interface 10.
- Switch B has a route to interface Loopback 1 (1.1.1.9/32) on Switch A, with the output interface VLAN-interface 12.
- Switch D has a route to 1.1.1.9/32, with the output interface VLAN-interface 10, and a route to 2.2.2.9/32, with the output interface VLAN-interface 12.

Configure the following:

- Configure a static route to subnet 120.1.1.0/24 on Switch A.
- Configure a static route to subnet 121.1.1.0/24 on Switch B.
- Enable BFD for both routes.

- Configure a static route to subnet 120.1.1.0/24 and a static route to subnet 121.1.1.0/24 on both Switch C and Switch D.

When the link between Switch A and Switch B through Switch D fails, BFD can detect the failure immediately. Switch A then communicates with Switch B through Switch C.

Figure 3 Network diagram

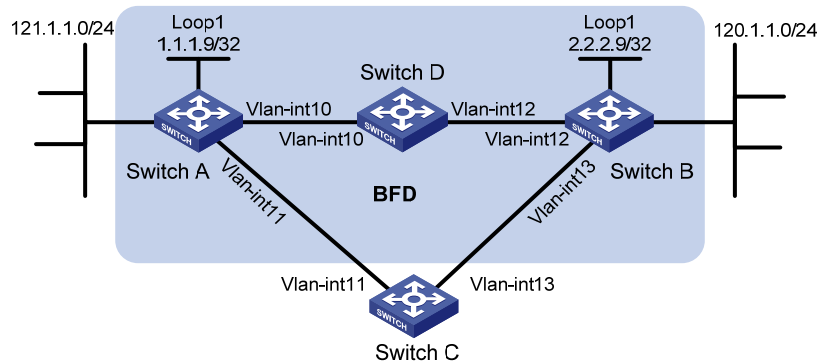


Table 4 Interface and IP address assignment

Device	Interface	IP address
Switch A	VLAN-interface 10	12.1.1.1/24
Switch A	VLAN-interface 11	10.1.1.102/24
Switch A	Loopback 1	1.1.1.9/32
Switch B	VLAN-interface 12	11.1.1.1/24
Switch B	VLAN-interface 13	13.1.1.1/24
Switch B	Loopback 1	2.2.2.9/32
Switch C	VLAN-interface 11	10.1.1.100/24
Switch C	VLAN-interface 13	13.1.1.2/24
Switch D	VLAN-interface 10	12.1.1.2/24
Switch D	VLAN-interface 12	11.1.1.2/24

Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)

2. Configure static routes and BFD:

Configure static routes on Switch A and enable BFD control mode for the static route that traverses Switch D.

```
<SwitchA> system-view
[SwitchA] bfd multi-hop min-transmit-interval 500
[SwitchA] bfd multi-hop min-receive-interval 500
[SwitchA] bfd multi-hop detect-multiplier 9
[SwitchA] ip route-static 120.1.1.0 24 2.2.2.9 bfd control-packet bfd-source 1.1.1.9
[SwitchA] ip route-static 120.1.1.0 24 vlan-interface 11 10.1.1.100 preference 65
[SwitchA] quit
```

Configure static routes on Switch B and enable BFD control mode for the static route that traverses Switch D.

```
<SwitchB> system-view
```

```
[SwitchB] bfd multi-hop min-transmit-interval 500
[SwitchB] bfd multi-hop min-receive-interval 500
[SwitchB] bfd multi-hop detect-multiplier 9
[SwitchB] ip route-static 121.1.1.0 24 1.1.1.9 bfd control-packet bfd-source 2.2.2.9
[SwitchB] ip route-static 121.1.1.0 24 vlan-interface 13 13.1.1.2 preference 65
[SwitchB] quit
```

Configure static routes on Switch C.

```
<SwitchC> system-view
[SwitchC] ip route-static 120.1.1.0 24 13.1.1.1
[SwitchC] ip route-static 121.1.1.0 24 10.1.1.102
```

Configure static routes on Switch D.

```
<SwitchD> system-view
[SwitchD] ip route-static 120.1.1.0 24 11.1.1.1
[SwitchD] ip route-static 121.1.1.0 24 12.1.1.1
```

Verifying the configuration

Display BFD sessions on Switch A.

```
<SwitchA> display bfd session
```

```
Total Session Num: 1      Up Session Num: 1      Init Mode: Active
```

```
IPv4 Session Working Under Ctrl Mode:
```

LD/RD	SourceAddr	DestAddr	State	Holdtime	Interface
4/7	1.1.1.9	2.2.2.9	Up	2000ms	N/A

The output shows that the BFD session has been created.

Display the static routes on Switch A.

```
<SwitchA> display ip routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
120.1.1.0/24	Static	60	0	12.1.1.2	Vlan10

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

The output shows that Switch A communicates with Switch B through VLAN-interface 10. Then the link over VLAN-interface 10 fails.

Display static routes on Switch A.

```
<SwitchA> display ip routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
120.1.1.0/24	Static	65	0	10.1.1.100	vlan11

Static Routing table Status : <Inactive>

Summary Count : 0

The output shows that Switch A communicates with Switch B through VLAN-interface 11.

Configuring a default route

A default route is used to forward packets that do not match any specific routing entry in the routing table. Without a default route, packets that do not match any routing entries are discarded.

A default route can be configured in either of the following ways:

- The network administrator can configure a default route with both destination and mask being 0.0.0.0. For more information, see "[Configuring a static route](#)."
- Some dynamic routing protocols, such as RIP, can generate a default route. For example, an upstream router running RIP can generate a default route and advertise it to other routers. These routers install the default route with the next hop being the upstream router. For more information, see the respective chapters on these routing protocols in this configuration guide.

Configuring RIP

Overview

Routing Information Protocol (RIP) is a distance-vector IGP suited to small-sized networks. It employs UDP to exchange route information through port 520.

RIP uses a hop count to measure the distance to a destination. The hop count from a router to a directly connected network is 0. The hop count from a router to a directly connected router is 1. To limit convergence time, RIP restricts the metric range from 0 to 15. A destination with a metric value of 16 (or greater) is considered unreachable. For this reason, RIP is not suitable for large-sized networks.

RIP route entries

RIP stores routing entries in a database. Each routing entry contains the following elements:

- **Destination address**—IP address of a destination host or a network.
- **Next hop**—IP address of the next hop.
- **Egress interface**—Egress interface of the route.
- **Metric**—Cost from the local router to the destination.
- **Route time**—Time elapsed since the last update. The time is reset to 0 when the routing entry is updated.
- **Route tag**—Used for route control. For more information, see "Configuring routing policies."

Routing loop prevention

RIP uses the following mechanisms to prevent routing loops:

- **Counting to infinity**—A destination with a metric value of 16 is considered unreachable. When a routing loop occurs, the metric value of a route will increment to 16 to avoid endless looping.
- **Triggered updates**—RIP immediately advertises triggered updates for topology changes to reduce the possibility of routing loops and to speed up convergence.
- **Split horizon**—Disables RIP from sending routes through the interface where the routes were learned to prevent routing loops and save bandwidth.
- **Poison reverse**—Enables RIP to set the metric of routes received from a neighbor to 16 and sends these routes back to the neighbor. The neighbor can delete such information from its routing table to prevent routing loops.

RIP operation

RIP works as follows:

1. RIP sends request messages to neighboring routers. Neighboring routers return response messages that contain their routing tables.
2. RIP uses the received responses to update the local routing table and sends triggered update messages to its neighbors. All RIP routers on the network do this to learn latest routing information.
3. RIP periodically sends the local routing table to its neighbors. After a RIP neighbor receives the message, it updates its routing table, selects optimal routes, and sends an update to other neighbors. RIP ages routes to keep only valid routes.

RIP versions

There are two RIP versions, RIPv1 and RIPv2.

RIPv1 is a classful routing protocol. It advertises messages through broadcast only. RIPv1 messages do not carry mask information, so RIPv1 can only recognize natural networks such as Class A, B, and C. For this reason, RIPv1 does not support discontinuous subnets.

RIPv2 is a classless routing protocol. It has the following advantages over RIPv1:

- Supports route tags to implement flexible route control through routing policies.
- Supports masks, route summarization, and CIDR.
- Supports designated next hops to select the best ones on broadcast networks.
- Supports multicasting route updates so only RIPv2 routers can receive these updates to reduce resource consumption.
- Supports plain text authentication and MD5 authentication to enhance security.

RIPv2 supports two transmission modes: broadcast and multicast. Multicast is the default mode using 224.0.0.9 as the multicast address. An interface operating in RIPv2 broadcast mode can also receive RIPv1 messages.

Protocols and standards

- RFC 1058, *Routing Information Protocol*
- RFC 1723, *RIP Version 2 - Carrying Additional Information*
- RFC 1721, *RIP Version 2 Protocol Analysis*
- RFC 1722, *RIP Version 2 Protocol Applicability Statement*
- RFC 1724, *RIP Version 2 MIB Extension*
- RFC 2082, *RIPv2 MD5 Authentication*
- RFC 2091, *Triggered Extensions to RIP to Support Demand Circuits*
- RFC 2453, *RIP Version 2*

RIP configuration task list

Tasks at a glance
<p>Configuring basic RIP:</p> <ul style="list-style-type: none">• (Required.) Enabling RIP• (Optional.) Controlling RIP reception and advertisement on interfaces• (Optional.) Configuring a RIP version
<p>(Optional.) Configuring RIP route control:</p> <ul style="list-style-type: none">• Configuring an additional routing metric• Configuring RIPv2 route summarization• Disabling host route reception• Advertising a default route• Configuring received/redistributed route filtering• Configuring a preference for RIP• Configuring RIP route redistribution
<p>(Optional.) Tuning and optimizing RIP networks:</p> <ul style="list-style-type: none">• Configuring RIP timers

Tasks at a glance

- [Configuring split horizon and poison reverse](#)
- [Enabling zero field check on incoming RIPv1 messages](#)
- [Enabling source IP address check on incoming RIP updates](#)
- [Configuring RIPv2 message authentication](#)
- [Specifying a RIP neighbor](#)
- [Configuring RIP network management](#)
- [Configuring the RIP packet sending rate](#)
- [Setting the maximum length of RIP packets](#)

(Optional.) [Configuring RIP GR](#)

(Optional.) [Configuring BFD for RIP](#)

Configuring basic RIP

Before you configure basic RIP settings, complete the following tasks:

- Configure the link layer protocol.
- Configure IP addresses for interfaces to ensure IP connectivity between neighboring routers.

Enabling RIP

To enable multiple RIP processes on a router, you must specify an ID for each process. A RIP process ID has only local significance. Two RIP routers having different process IDs can also exchange RIP packets.

If you configure RIP settings in interface view before enabling RIP, the settings do not take effect until RIP is enabled. If a physical interface is attached to multiple networks, you cannot advertise these networks in different RIP processes. You cannot enable multiple RIP processes on a physical interface.

Enabling RIP on a network

You can enable RIP on a network and specify a wildcard mask for the network. After that, only the interface attached to the network runs RIP.

To enable RIP on a network:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enable RIP and enter RIP view.	rip [<i>process-id</i>]	By default, RIP is disabled.
3. Enable RIP on a network.	network <i>network-address</i> [<i>wildcard-mask</i>]	By default, RIP is disabled on a network. The network 0.0.0.0 command can enable RIP on all interfaces in a single process, but does not apply to multiple RIP processes.

Enabling RIP on an interface

Step	Command	Remarks
1. Enter system view.	system-view	N/A

Step	Command	Remarks
2. Enable RIP and enter RIP view.	rip [<i>process-id</i>]	By default, RIP is disabled.
3. Return to system view.	quit	N/A
4. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
5. Enable RIP on the interface.	rip process-id enable [exclude-subip]	By default, RIP is disabled on an interface.

Controlling RIP reception and advertisement on interfaces

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Disable an interface from sending RIP messages.	silent-interface { <i>interface-type</i> <i>interface-number</i> all }	By default, all RIP-enabled interfaces can send RIP messages. The disabled interface can still receive RIP messages and respond to unicast requests containing unknown ports.
4. Return to system view.	quit	N/A
5. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
6. Enable an interface to receive RIP messages.	rip input	By default, a RIP-enabled interface can receive RIP messages.
7. Enable an interface to send RIP messages.	rip output	By default, a RIP-enabled interface can send RIP messages.

Configuring a RIP version

You can configure a global RIP version in RIP view or an interface-specific RIP version in interface view.

An interface preferentially uses the interface-specific RIP version. If no interface-specific version is specified, the interface uses the global RIP version. If neither a global nor interface-specific RIP version is configured, the interface sends RIPv1 broadcasts and can receive the following:

- RIPv1 broadcasts and unicasts.
- RIPv2 broadcasts, multicasts, and unicasts.

To configure a RIP version:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Specify a global RIP version.	version { 1 2 }	By default, no global version is

Step	Command	Remarks
		specified. An interface sends RIPv1 broadcasts, and can receive RIPv1 broadcasts and unicasts, and RIPv2 broadcasts, multicasts, and unicasts.
4. Return to system view.	quit	N/A
5. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
6. Specify a RIP version for the interface.	rip version { 1 2 [broadcast multicast] }	By default, no interface-specific RIP version is specified. The interface sends RIPv1 broadcasts, and can receive RIPv1 broadcasts and unicasts, and RIPv2 broadcasts, multicasts, and unicasts.

Configuring RIP route control

Before you configure RIP route control, complete the following tasks:

- Configure IP addresses for interfaces to ensure IP connectivity between neighboring routers.
- Configure basic RIP.

Configuring an additional routing metric

An additional routing metric (hop count) can be added to the metric of an inbound or outbound RIP route.

An outbound additional metric is added to the metric of a sent route, and it does not change the route's metric in the routing table.

An inbound additional metric is added to the metric of a received route before the route is added into the routing table, and the route's metric is changed. If the sum of the additional metric and the original metric is greater than 16, the metric of the route is 16.

To configure additional routing metrics:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Specify an inbound additional routing metric.	rip metricin [route-policy <i>route-policy-name</i>] <i>value</i>	The default setting is 0.
4. Specify an outbound additional routing metric.	rip metricout [route-policy <i>route-policy-name</i>] <i>value</i>	The default setting is 1.

Configuring RIPv2 route summarization

Perform this task to summarize contiguous subnets into a summary network and sends the network to neighbors. The smallest metric among all summarized routes is used as the metric of the summary route.

Enabling RIPv2 automatic route summarization

Automatic summarization enables RIPv2 to generate a natural network for contiguous subnets. For example, suppose there are three subnet routes 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24. Automatic summarization automatically creates and advertises a summary route 10.0.0.0/8 instead of the more specific routes.

To enable RIPv2 automatic route summarization:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [process-id]	N/A
3. (Optional.) Enable RIPv2 automatic route summarization.	summary	By default, RIPv2 automatic route summarization is enabled. If subnets in the routing table are not contiguous, disable automatic route summarization to advertise more specific routes.

Advertising a summary route

Perform this task to manually configure a summary route.

For example, suppose contiguous subnets routes 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 exist in the routing table. You can create a summary route 10.1.0.0/16 on VLAN-interface 1 to advertise the summary route instead of the more specific routes.

To configure a summary route:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [process-id]	N/A
3. Disable RIPv2 automatic route summarization.	undo summary	By default, RIPv2 automatic route summarization is enabled.
4. Return to system view.	quit	N/A
5. Enter interface view.	interface interface-type interface-number	N/A
6. Configure a summary route.	rip summary-address ip-address { mask-length mask }	By default, no summary route is configured.

Disabling host route reception

Perform this task to disable RIPv2 from receiving host routes from the same network to save network resources. This feature does not apply to RIPv1.

To disable RIP from receiving host routes:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [process-id]	N/A
3. Disable RIP from receiving host routes.	undo host-route	By default, RIP receives host routes.

Advertising a default route

You can advertise a default route on all RIP interfaces in RIP view or on a specific RIP interface in interface view. The interface view setting takes precedence over the RIP view settings.

To disable an interface from advertising a default route, use the **rip default-route no-originate** command on the interface.

To configure RIP to advertise a default route:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Enable RIP to advertise a default route.	default-route { only originate } [<i>cost cost</i>]	By default, RIP does not advertise a default route.
4. Return to system view.	quit	N/A
5. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
6. Configure the RIP interface to advertise a default route.	rip default-route { { only originate } [<i>cost cost</i>] no-originate }	By default, a RIP interface can advertise a default route if the RIP process is enabled to advertise a default route.

NOTE:

The router enabled to advertise a default route does not accept default routes from RIP neighbors.

Configuring received/redistributed route filtering

Perform this task to filter received and redistributed routes by using a filtering policy.

To configure route filtering:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Configure the filtering of received routes.	filter-policy { <i>acl-number</i> gateway <i>prefix-list-name</i> prefix-list <i>prefix-list-name</i> [gateway <i>prefix-list-name</i>] } import [<i>interface-type interface-number</i>]	By default, the filtering of received routes is not configured. This command filters received routes. Filtered routes are not installed into the routing table or advertised to neighbors.
4. Configure the filtering of redistributed routes.	filter-policy { <i>acl-number</i> prefix-list <i>prefix-list-name</i> } export [<i>protocol</i> [<i>process-id</i>] <i>interface-type interface-number</i>]	By default, the filtering of redistributed routes is not configured. This command filters redistributed routes, including routes redistributed with the import-route command.

Configuring a preference for RIP

If multiple IGPs find routes to the same destination, the route found by the IGP that has the highest priority is selected as the optimal route. Perform this task to assign a preference to RIP. The smaller the preference value, the higher the priority.

To configure a preference for RIP:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Configure a preference for RIP.	preference [route-policy <i>route-policy-name</i>] <i>value</i>	The default setting is 100.

Configuring RIP route redistribution

Perform this task to configure RIP to redistribute routes from other routing protocols, including static, and direct.

To configure RIP route redistribution:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Redistribute routes from another routing protocol.	import-route <i>protocol</i> [<i>process-id</i> all-processes] [cost <i>cost</i> route-policy <i>route-policy-name</i> tag <i>tag</i>] *	By default, RIP route redistribution is disabled. This command can redistribute only active routes. To view active routes, use the display ip routing-table protocol command.
4. (Optional.) Configure a default cost for redistributed routes.	default cost <i>value</i>	The default setting is 0.

Tuning and optimizing RIP networks

Configuration prerequisites

Before you tune and optimize RIP networks, complete the following tasks:

- Configure IP addresses for interfaces to ensure IP connectivity between neighboring nodes.
- Configure basic RIP.

Configuring RIP timers

You can change the RIP network convergence speed by adjusting the following RIP timers:

- **Update timer**—Specifies the interval between route updates.

- **Timeout timer**—Specifies the route aging time. If no update for a route is received within the aging time, the metric of the route is set to 16.
- **Suppress timer**—Specifies how long a RIP route stays in suppressed state. When the metric of a route is 16, the route enters the suppressed state. A suppressed route can be replaced by an updated route that is received from the same neighbor before the suppress timer expires and has a metric less than 16.
- **Garbage-collect timer**—Specifies the interval from when the metric of a route becomes 16 to when it is deleted from the routing table. RIP advertises the route with a metric of 16. If no update is announced for that route before the garbage-collect timer expires, the route is deleted from the routing table.

! **IMPORTANT:**

To avoid unnecessary traffic or route flapping, configure identical RIP timer settings on RIP routers.

To configure RIP timers:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Configure RIP timers.	timers { garbage-collect <i>garbage-collect-value</i> suppress <i>suppress-value</i> timeout <i>timeout-value</i> update <i>update-value</i> } *	By default: <ul style="list-style-type: none"> • The garbage-collect timer is 120 seconds. • The suppress timer is 120 seconds. • The timeout timer is 180 seconds. • The update timer is 30 seconds.

Configuring split horizon and poison reverse

The split horizon and poison reverse functions can prevent routing loops.

If both split horizon and poison reverse are configured, only the poison reverse function takes effect.

Enabling split horizon

Split horizon disables RIP from sending routes through the interface where the routes were learned to prevent routing loops between adjacent routers.

To enable split horizon:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Enable split horizon.	rip split-horizon	By default, split horizon is enabled.

Enabling poison reverse

Poison reverse allows RIP to send routes through the interface where the routes were learned. The metric of these routes is always set to 16 (unreachable) to avoid routing loops between neighbors.

To enable poison reverse:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Enable poison reverse.	rip poison-reverse	By default, poison reverse is disabled.

Enabling zero field check on incoming RIPv1 messages

Some fields in the RIPv1 message must be set to zero. These fields are called "zero fields." You can enable zero field check on incoming RIPv1 messages. If a zero field of a message contains a non-zero value, RIP does not process the message. If you are certain that all messages are trustworthy, disable zero field check to save CPU resources.

This feature does not apply to RIPv2 packets, because they have no zero fields.

To enable zero field check on incoming RIPv1 messages:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Enable zero field check on incoming RIPv1 messages.	checkzero	The default setting is enabled.

Enabling source IP address check on incoming RIP updates

Perform this task to enable source IP address check on incoming RIP updates.

Upon receiving a message on an Ethernet interface, RIP compares the source IP address of the message with the IP address of the interface. If they are not in the same network segment, RIP discards the message.

Upon receiving a message on a PPP interface, RIP checks whether the source address of the message is the IP address of the peer interface. If not, RIP discards the message.

To enable source IP address check on incoming RIP updates:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Enable source IP address check on incoming RIP messages.	validate-source-address	By default, this function is enabled.

Configuring RIPv2 message authentication

Perform this task to enable authentication on RIPv2 messages. This feature does not apply to RIPv1 because RIPv1 does not support authentication. Although you can specify an authentication mode for RIPv1 in interface view, the configuration does not take effect.

RIPv2 supports two authentication modes: simple authentication and MD5 authentication.

To configure RIPv2 message authentication:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type interface-number</i>	N/A
3. Configure RIPv2 authentication.	rip authentication-mode { md5 { rfc2082 { cipher <i>cipher-string</i> plain <i>plain-string</i> } <i>key-id</i> rfc2453 { cipher <i>cipher-string</i> plain <i>plain-string</i> } } simple { cipher <i>cipher-string</i> plain <i>plain-string</i> } }	By default, RIPv2 authentication is not configured.

Specifying a RIP neighbor

Typically RIP messages are sent in broadcast or multicast. To enable RIP on a link that does not support broadcast or multicast, you must manually specify RIP neighbors.

Follow these guidelines when you specify a RIP neighbor:

- Do not use the **peer** *ip-address* command when the neighbor is directly connected. Otherwise, the neighbor might receive both unicast and multicast (or broadcast) messages of the same routing information.
- If the specified neighbor is not directly connected, disable source address check on incoming updates.

To specify a RIP neighbor:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Specify a RIP neighbor.	peer <i>ip-address</i>	By default, RIP does not unicast updates to any peer.
4. Disable source IP address check on inbound RIP updates	undo validate-source-address	By default, source IP address check on inbound RIP updates is enabled.

Configuring RIP network management

You can use network management software to manage the RIP process to which MIB is bound.

To configure RIP network management:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Bind MIB to a RIP process.	rip mib-binding <i>process-id</i>	By default, MIB is bound to the RIP process with the smallest process ID.

Configuring the RIP packet sending rate

Perform this task to specify the interval for sending RIP packets and the maximum number of RIP packets that can be sent at each interval. This feature can avoid excessive RIP packets from affecting system performance and consuming too much bandwidth.

To configure the RIP packet sending rate:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Specify the interval for sending RIP packets and the maximum number of RIP packets that can be sent at each interval.	output-delay <i>time count count</i>	By default, an interface sends up to three RIP packets every 20 milliseconds.

Setting the maximum length of RIP packets

NOTE:

The supported maximum length of RIP packets varies by vendor. Use this feature with caution to avoid compatibility issues.

The packet length of RIP packets determines how many routes can be carried in a RIP packet. Set the maximum length of RIP packets to make good use of link bandwidth.

When authentication is enabled, follow these guidelines to ensure packet forwarding:

- For simple authentication, the maximum length of RIP packets must be no less than 52 bytes.
- For MD5 authentication (with packet format defined in RFC 2453), the maximum length of RIP packets must be no less than 56 bytes.
- For MD5 authentication (with packet format defined in RFC 2082), the maximum length of RIP packets must be no less than 72 bytes.

To set the maximum length of RIP packets:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type interface-number</i>	N/A
3. Set the maximum length of RIP packets.	rip max-packet-length <i>value</i>	By default, the maximum length of RIP packets is 512 bytes.

Configuring RIP GR

GR ensures forwarding continuity when a routing protocol restarts or an active/standby switchover occurs.

Two routers are required to complete a GR process. The following are router roles in a GR process.

- **GR restarter**—Graceful restarting router. It must have GR capability.

- **GR helper**—A neighbor of the GR restarter. It helps the GR restarter to complete the GR process.

After RIP restarts on a router, the router must learn RIP routes again and update its FIB table, which causes network disconnections and route reconvergence.

With the GR feature, the restarting router (known as the "GR restarter") can notify the event to its GR capable neighbors. GR capable neighbors (known as "GR helpers") maintain their adjacencies with the router within a GR interval. During this process, the FIB table of the router does not change. After the restart, the router contacts its neighbors to retrieve its FIB.

By default, a RIP-enabled device acts as the GR helper. Perform this task on the GR restarter.

To configure GR on the GR restarter:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Enable GR for RIP.	graceful-restart	By default, RIP GR is disabled.

Configuring BFD for RIP

RIP detects route failures by periodically sending requests. If it receives no response for a route within a certain time, RIP considers the route unreachable. To speed up convergence, perform this task to enable BFD for RIP. For more information about BFD, see *High Availability Configuration Guide*.

RIP supports the following BFD detection modes:

- **Single-hop echo detection**—Detection mode for a direct neighbor. In this mode, a BFD session is established only when the directly connected neighbor has route information to send.
- **Single-hop echo detection for a specific destination**—In this mode, a BFD session is established to the specified RIP neighbor when RIP is enabled on the local interface.
- **Bidirectional control detection**—Detection mode for an indirect neighbor. In this mode, a BFD session is established only when both ends have routes to send and BFD is enabled on the receiving interface.

Configuring single-hop echo detection (for a directly connected RIP neighbor)

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure the source IP address of BFD echo packets.	bfd echo-source-ip <i>ip-address</i>	By default, the source IP address of BFD echo packets is not configured.
3. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
4. Enable BFD for RIP.	rip bfd enable	By default, BFD for RIP is disabled.

Configuring single-hop echo detection (for a specific destination)

When a unidirectional link occurs between the local device and a specific neighbor, BFD can detect the failure. The local device will not receive or send any RIP packets through the interface connected to the neighbor to improve convergence speed. When the link recovers, the interface can send RIP packets again.

This feature applies to RIP neighbors that are directly connected.

To configure BFD for RIP (single hop echo detection for a specific destination):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure the source IP address of BFD echo packets.	bfd echo-source-ip <i>ip-address</i>	By default, no source IP address is configured for BFD echo packets.
3. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
4. Enable BFD for RIP.	rip bfd enable destination <i>ip-address</i>	By default, BFD for RIP is disabled.

Configuring bidirectional control detection

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIP view.	rip [<i>process-id</i>]	N/A
3. Specify a RIP neighbor.	peer <i>ip-address</i>	By default, RIP does not unicast updates to any peer. Because the undo peer command does not remove the neighbor relationship immediately, executing the command cannot bring down the BFD session immediately.
4. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
5. Enable BFD on the RIP interface.	rip bfd enable	By default, BFD is disabled on a RIP interface.

Displaying and maintaining RIP

Execute **display** commands in any view and execute **reset** commands in user view.

Task	Command
Display RIP current status and configuration information.	display rip [<i>process-id</i>]

Task	Command
Display active routes in RIP database.	display rip process-id database [ip-address { mask-length mask }]
Display RIP interface information.	display rip process-id interface [interface-type interface-number]
Display routing information about a specified RIP process.	display rip process-id route [ip-address { mask-length mask } [verbose] peer ip-address statistics]
Reset a RIP process.	reset rip process-id process
Clear the statistics for a RIP process.	reset rip process-id statistics

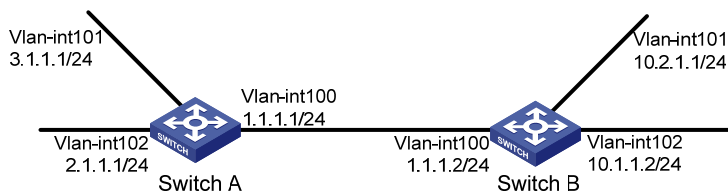
RIP configuration examples

Basic RIP configuration example

Network requirements

As shown in [Figure 4](#), enable RIPv2 on all interfaces on Switch A and Switch B. Configure Switch B to not advertise route 10.2.1.0/24 to Switch A, and to accept only route 2.1.1.0/24 from Switch A.

Figure 4 Network diagram



Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)
2. Configure basic RIP:

Enable RIP on the specified networks on Switch A.

```
<SwitchA> system-view
[SwitchA] rip
[SwitchA-rip-1] network 1.0.0.0
[SwitchA-rip-1] network 2.0.0.0
[SwitchA-rip-1] network 3.0.0.0
[SwitchA-rip-1] quit
```

Enable RIP on the specified interfaces on Switch B.

```
<SwitchB> system-view
[SwitchB] rip
[SwitchB-rip-1] quit
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] rip 1 enable
[SwitchB-Vlan-interface100] quit
[SwitchB] interface vlan-interface 101
[SwitchB-Vlan-interface101] rip 1 enable
[SwitchB-Vlan-interface101] quit
```

```
[SwitchB] interface vlan-interface 102
[SwitchB-Vlan-interface102] rip 1 enable
[SwitchB-Vlan-interface102] quit
```

Display the RIP routing table of Switch A.

```
[SwitchA] display rip 1 route
```

```
Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
              O - Optimal, F - Flush to RIB
```

```
-----
Peer 1.1.1.2 on Vlan-interface100
  Destination/Mask    Nexthop          Cost    Tag    Flags  Sec
  10.0.0.0/8         1.1.1.2          1       0      RAOF   11
Local route
  Destination/Mask    Nexthop          Cost    Tag    Flags  Sec
  1.1.1.0/24         0.0.0.0          0       0      RDOF   -
  2.1.1.0/24         0.0.0.0          0       0      RDOF   -
  3.1.1.0/24         0.0.0.0          0       0      RDOF   -
```

The output shows that RIPv1 uses a natural mask.

3. Configure a RIP version:

Configure RIPv2 on Switch A.

```
[SwitchA] rip
[SwitchA-rip-1] version 2
[SwitchA-rip-1] undo summary
[SwitchA-rip-1] quit
```

Configure RIPv2 on Switch B.

```
[SwitchB] rip
[SwitchB-rip-1] version 2
[SwitchB-rip-1] undo summary
[SwitchB-rip-1] quit
```

Display the RIP routing table on Switch A.

```
[SwitchA] display rip 1 route
```

```
Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
              O - Optimal, F - Flush to RIB
```

```
-----
Peer 1.1.1.2 on Vlan-interface100
  Destination/Mask    Nexthop          Cost    Tag    Flags  Sec
  10.0.0.0/8         1.1.1.2          1       0      RAOF   50
Local route
  Destination/Mask    Nexthop          Cost    Tag    Flags  Sec
  1.1.1.0/24         0.0.0.0          0       0      RDOF   -
  2.1.1.0/24         0.0.0.0          0       0      RDOF   -
  3.1.1.0/24         0.0.0.0          0       0      RDOF   -
```

The output shows that RIPv2 uses classless subnet masks.

NOTE:

After RIPv2 is configured, RIPv1 routes might still exist in the routing table until they are aged out.

Display the RIP routing table on Switch B.

```
Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
              O - Optimal, F - Flush to RIB
-----
Peer 1.1.1.1 on Vlan-interface100
  Destination/Mask    Nexthop          Cost    Tag    Flags    Sec
  2.1.1.0/24         1.1.1.1         1       0      RAOF     19
  3.1.1.0/24         1.1.1.1         1       0      RAOF     19
Local route
  Destination/Mask    Nexthop          Cost    Tag    Flags    Sec
  1.1.1.0/24         0.0.0.0         0       0      RDOF     -
  10.1.1.0/24        0.0.0.0         0       0      RDOF     -
  10.2.1.0/24        0.0.0.0         0       0      RDOF     -
```

4. Configure route filtering:**# Reference IP prefix lists on Switch B to filter received and redistributed routes.**

```
[SwitchB] ip prefix-list aaa index 10 permit 2.1.1.0 24
[SwitchB] ip prefix-list bbb index 10 permit 10.1.1.0 24
[SwitchB] rip 1
[SwitchB-rip-1] filter-policy prefix-list aaa import
[SwitchB-rip-1] filter-policy prefix-list bbb export
[SwitchB-rip-1] quit
```

Display the RIP routing table on Switch A.

```
[SwitchA] display rip 100 route
Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
              O - Optimal, F - Flush to RIB
-----
Peer 1.1.1.2 on Vlan-interface100
  Destination/Mask    Nexthop          Cost    Tag    Flags    Sec
  10.1.1.0/24        1.1.1.2         1       0      RAOF     19
Local route
  Destination/Mask    Nexthop          Cost    Tag    Flags    Sec
  1.1.1.0/24         0.0.0.0         0       0      RDOF     -
  2.1.1.0/24         0.0.0.0         0       0      RDOF     -
  3.1.1.0/24         0.0.0.0         0       0      RDOF     -
```

Display the RIP routing table on Switch B.

```
[SwitchB] display rip 1 route
Route Flags: R - RIP
              A - Aging, S - Suppressed, G - Garbage-collect
              O - Optimal, F - Flush to RIB
-----
Peer 1.1.1.1 on Vlan-interface100
  Destination/Mask    Nexthop          Cost    Tag    Flags    Sec
```

2.1.1.0/24	1.1.1.1	1	0	RAOF	19
Local route					
Destination/Mask	NextHop	Cost	Tag	Flags	Sec
1.1.1.0/24	0.0.0.0	0	0	RDOF	-
10.1.1.0/24	0.0.0.0	0	0	RDOF	-
10.2.1.0/24	0.0.0.0	0	0	RDOF	-

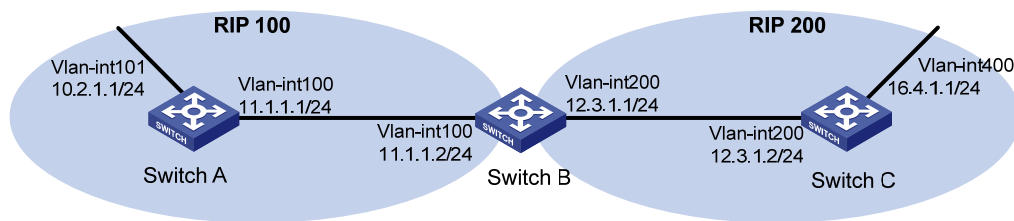
RIP route redistribution configuration example

Network requirements

As shown in [Figure 5](#), Switch B communicates with Switch A through RIP 100 and with Switch C through RIP 200.

Configure RIP 200 to redistribute direct routes and routes from RIP 100 on Switch B so Switch C can learn routes destined for 10.2.1.0/24 and 11.1.1.0/24. Switch A cannot learn routes destined for 12.3.1.0/24 and 16.4.1.0/24.

Figure 5 Network diagram



Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)
2. Configure basic RIP:

Enable RIP 100, and configure RIPv2 on Switch A.

```
<SwitchA> system-view
[SwitchA] rip 100
[SwitchA-rip-100] network 10.0.0.0
[SwitchA-rip-100] network 11.0.0.0
[SwitchA-rip-100] version 2
[SwitchA-rip-100] undo summary
[SwitchA-rip-100] quit
```

Enable RIP 100 and RIP 200, and configure RIPv2 on Switch B.

```
<SwitchB> system-view
[SwitchB] rip 100
[SwitchB-rip-100] network 11.0.0.0
[SwitchB-rip-100] version 2
[SwitchB-rip-100] undo summary
[SwitchB-rip-100] quit
[SwitchB] rip 200
[SwitchB-rip-200] network 12.0.0.0
[SwitchB-rip-200] version 2
[SwitchB-rip-200] undo summary
[SwitchB-rip-200] quit
```

Enable RIP 200, and configure RIPv2 on Switch C.

```

<SwitchC> system-view
[SwitchC] rip 200
[SwitchC-rip-200] network 12.0.0.0
[SwitchC-rip-200] network 16.0.0.0
[SwitchC-rip-200] version 2
[SwitchC-rip-200] undo summary
[SwitchC-rip-200] quit

```

Display the IP routing table on Switch C.

```
[SwitchC] display ip routing-table
```

```
Destinations : 13          Routes : 13
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
12.3.1.0/24	Direct	0	0	12.3.1.2	Vlan200
12.3.1.0/32	Direct	0	0	12.3.1.2	Vlan200
12.3.1.2/32	Direct	0	0	127.0.0.1	InLoop0
12.3.1.255/32	Direct	0	0	12.3.1.2	Vlan200
16.4.1.0/24	Direct	0	0	16.4.1.1	Vlan400
16.4.1.0/32	Direct	0	0	16.4.1.1	Vlan400
16.4.1.1/32	Direct	0	0	127.0.0.1	InLoop0
16.4.1.255/32	Direct	0	0	16.4.1.1	Vlan400
127.0.0.0/8	Direct	0	0	127.0.0.1	InLoop0
127.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
127.0.0.1/32	Direct	0	0	127.0.0.1	InLoop0
127.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0

3. Configure route redistribution:

Configure RIP 200 to redistribute routes from RIP 100 and direct routes on Switch B.

```

[SwitchB] rip 200
[SwitchB-rip-200] import-route rip 100
[SwitchB-rip-200] import-route direct
[SwitchB-rip-200] quit

```

Display the IP routing table on Switch C.

```
[SwitchC] display ip routing-table
```

```
Destinations : 15          Routes : 15
```

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
10.2.1.0/24	RIP	100	1	12.3.1.1	Vlan200
11.1.1.0/24	RIP	100	1	12.3.1.1	Vlan200
12.3.1.0/24	Direct	0	0	12.3.1.2	Vlan200
12.3.1.0/32	Direct	0	0	12.3.1.2	Vlan200
12.3.1.2/32	Direct	0	0	127.0.0.1	InLoop0
12.3.1.255/32	Direct	0	0	12.3.1.2	Vlan200
16.4.1.0/24	Direct	0	0	16.4.1.1	Vlan400
16.4.1.0/32	Direct	0	0	16.4.1.1	Vlan400
16.4.1.1/32	Direct	0	0	127.0.0.1	InLoop0

16.4.1.255/32	Direct	0	0	16.4.1.1	Vlan400
127.0.0.0/8	Direct	0	0	127.0.0.1	InLoop0
127.0.0.0/32	Direct	0	0	127.0.0.1	InLoop0
127.0.0.1/32	Direct	0	0	127.0.0.1	InLoop0
127.255.255.255/32	Direct	0	0	127.0.0.1	InLoop0

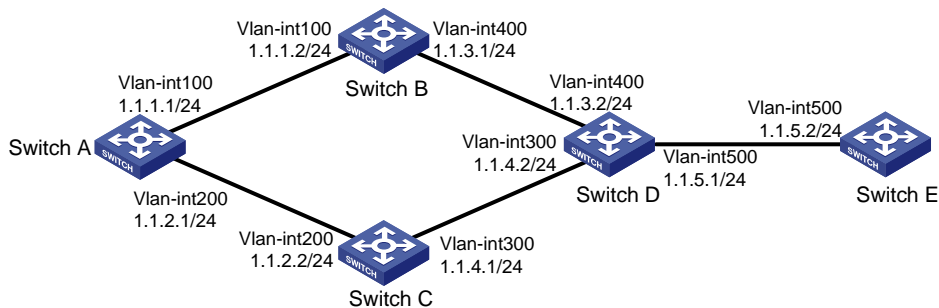
RIP interface additional metric configuration example

Network requirements

As shown in [Figure 6](#), run RIPv2 on all the interfaces of Switch A, Switch B, Switch C, Switch D, and Switch E.

Switch A has two links to Switch D. The link from Switch B to Switch D is more stable than that from Switch C to Switch D. Configure an additional metric for RIP routes received from VLAN-interface 200 on Switch A so Switch A prefers route 1.1.5.0/24 learned from Switch B.

Figure 6 Network diagram



Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)

2. Configure basic RIP:

Configure Switch A.

```

<SwitchA> system-view
[SwitchA] rip 1
[SwitchA-rip-1] network 1.0.0.0
[SwitchA-rip-1] version 2
[SwitchA-rip-1] undo summary
[SwitchA-rip-1] quit
  
```

Configure Switch B.

```

<SwitchB> system-view
[SwitchB] rip 1
[SwitchB-rip-1] network 1.0.0.0
[SwitchB-rip-1] version 2
[SwitchB-rip-1] undo summary
  
```

Configure Switch C.

```

<SwitchC> system-view
[SwitchC] rip 1
[SwitchC-rip-1] network 1.0.0.0
[SwitchC-rip-1] version 2
[SwitchC-rip-1] undo summary
  
```

Configure Switch D.

```
<SwitchD> system-view
[SwitchD] rip 1
[SwitchD-rip-1] network 1.0.0.0
[SwitchD-rip-1] version 2
[SwitchD-rip-1] undo summary
```

Configure Switch E.

```
<SwitchE> system-view
[SwitchE] rip 1
[SwitchE-rip-1] network 1.0.0.0
[SwitchE-rip-1] version 2
[SwitchE-rip-1] undo summary
```

Display all active routes in the RIP database on Switch A.

```
[SwitchA] display rip 1 database
 1.0.0.0/8, auto-summary
   1.1.1.0/24, cost 0, nexthop 1.1.1.1, RIP-interface
   1.1.2.0/24, cost 0, nexthop 1.1.2.1, RIP-interface
   1.1.3.0/24, cost 1, nexthop 1.1.1.2
   1.1.4.0/24, cost 1, nexthop 1.1.2.2
   1.1.5.0/24, cost 2, nexthop 1.1.1.2
   1.1.5.0/24, cost 2, nexthop 1.1.2.2
```

The output shows two RIP routes destined for network 1.1.5.0/24. The next hops of the routes are Switch B (1.1.1.2) and Switch C (1.1.2.2). The cost of the routes is 2.

3. Configure an additional metric for a RIP interface:

Configure an inbound additional metric of 3 for RIP-enabled interface VLAN-interface 200 on Switch A.

```
[SwitchA] interface vlan-interface 200
[SwitchA-Vlan-interface200] rip metricin 3
```

Display all active routes in the RIP database on Switch A.

```
[SwitchA-Vlan-interface200] display rip 1 database
 1.0.0.0/8, auto-summary
   1.1.1.0/24, cost 0, nexthop 1.1.1.1, RIP-interface
   1.1.2.0/24, cost 0, nexthop 1.1.2.1, RIP-interface
   1.1.3.0/24, cost 1, nexthop 1.1.1.2
   1.1.4.0/24, cost 2, nexthop 1.1.1.2
   1.1.5.0/24, cost 2, nexthop 1.1.1.2
```

The output shows that only one RIP route reaches network 1.1.5.0/24, with the next hop as Switch B (1.1.1.2) and a cost of 2.

BFD for RIP configuration example (single-hop echo detection for a directly connected neighbor)

Network requirements

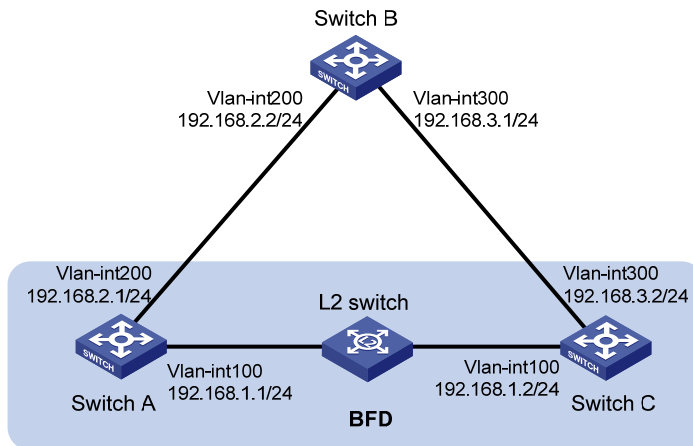
As shown in [Figure 7](#), VLAN-interface 100 of Switch A and Switch C runs RIP process 1. VLAN-interface 200 of Switch A runs RIP process 2. VLAN-interface 300 of Switch C and VLAN-interface 200 and VLAN-interface 300 of Switch B run RIP process 1.

- Configure a static route destined for 100.1.1.1/24 and enable static route redistribution into RIP on Switch C. This allows Switch A to learn two routes destined for 100.1.1.1/24 through

VLAN-interface 100 and VLAN-interface 200 respectively, and uses the one through VLAN-interface 100.

- Enable BFD for RIP on VLAN-interface 100 of Switch A. When the link over VLAN-interface 100 fails, BFD can quickly detect the failure and notify it to RIP. RIP deletes the neighbor relationship and route information learned on VLAN-interface 100. It uses the route destined for 100.1.1.1 24 through VLAN-interface 200.

Figure 7 Network diagram



Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)
2. Configure basic RIP:

Configure Switch A.

```
<SwitchA> system-view
[SwitchA] rip 1
[SwitchA-rip-1] version 2
[SwitchA-rip-1] undo summary
[SwitchA-rip-1] network 192.168.1.0
[SwitchA-rip-1] quit
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] rip bfd enable
[SwitchA-Vlan-interface100] quit
[SwitchA] rip 2
[SwitchA-rip-2] version 2
[SwitchA-rip-2] undo summary
[SwitchA-rip-2] network 192.168.2.0
[SwitchA-rip-2] quit
```

Configure Switch B.

```
<SwitchB> system-view
[SwitchB] rip 1
[SwitchB-rip-1] version 2
[SwitchB-rip-1] undo summary
[SwitchB-rip-1] network 192.168.2.0
[SwitchB-rip-1] network 192.168.3.0
[SwitchB-rip-1] quit
```

Configure Switch C.

```

<SwitchC> system-view
[SwitchC] rip 1
[SwitchC-rip-1] version 2
[SwitchC-rip-1] undo summary
[SwitchC-rip-1] network 192.168.1.0
[SwitchC-rip-1] network 192.168.3.0
[SwitchC-rip-1] import-route static
[SwitchC-rip-1] quit

```

3. Configure BFD parameters on VLAN-interface 100 of Switch A.

```

[SwitchA] bfd session init-mode active
[SwitchA] bfd echo-source-ip 11.11.11.11
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] bfd min-transmit-interval 500
[SwitchA-Vlan-interface100] bfd min-receive-interval 500
[SwitchA-Vlan-interface100] bfd detect-multiplier 7
[SwitchA-Vlan-interface100] quit
[SwitchA] quit

```

4. Configure a static route on Switch C.

```

[SwitchC] ip route-static 120.1.1.1 24 null 0

```

Verifying the configuration

Display the BFD session information on Switch A.

```

<SwitchA> display bfd session

```

```

Total Session Num: 1      Up Session Num: 1      Init Mode: Active

```

```

IPv4 Session Working Under Echo Mode:

```

LD	SourceAddr	DestAddr	State	Holdtime	Interface
4	192.168.1.1	192.168.1.2	Up	2000ms	Vlan100

Display RIP routes destined for 120.1.1.0/24 on Switch A.

```

<SwitchA> display ip routing-table 120.1.1.0 24 verbose

```

```

Summary Count : 1

```

```

Destination: 120.1.1.0/24

```

```

  Protocol: RIP          Process ID: 1
  SubProtID: 0x1        Age: 04h20m37s
  Cost: 1               Preference: 100
  Tag: 0                State: Active Adv
  OrigTblID: 0x0        OrigVrf: default-vrf
  TableID: 0x2          OrigAs: 0
  NBRID: 0x26000002     LastAs: 0
  AttrID: 0xffffffff    Neighbor: 192.168.1.2
  Flags: 0x1008c        OrigNextHop: 192.168.1.2
  Label: NULL           RealNextHop: 192.168.1.2
  BkLabel: NULL         BkNextHop: N/A
  Tunnel ID: Invalid    Interface: Vlan-interface100

```

BkTunnel ID: Invalid BkInterface: N/A

The output shows that Switch A communicates with Switch C through VLAN-interface 100. Then the link over VLAN-interface 100 fails.

Display RIP routes destined for 120.1.1.0/24 on Switch A.

```
<SwitchA> display ip routing-table 120.1.1.0 24 verbose
```

```
Summary Count : 1
```

```
Destination: 120.1.1.0/24
```

```
Protocol: RIP                    Process ID: 2
SubProtID: 0x1                    Age: 04h20m37s
Cost: 1                            Preference: 100
Tag: 0                             State: Active Adv
OrigTblID: 0x0                    OrigVrf: default-vrf
TableID: 0x2                      OrigAs: 0
NBRID: 0x26000002                LastAs: 0
AttrID: 0xffffffff               Neighbor: 192.168.2.2
Flags: 0x1008c                   OrigNextHop: 192.168.2.2
Label: NULL                       RealNextHop: 192.168.2.2
BkLabel: NULL                     BkNextHop: N/A
Tunnel ID: Invalid                Interface: Vlan-interface200
BkTunnel ID: Invalid              BkInterface: N/A
```

The output shows that Switch A communicates with Switch C through VLAN-interface 200.

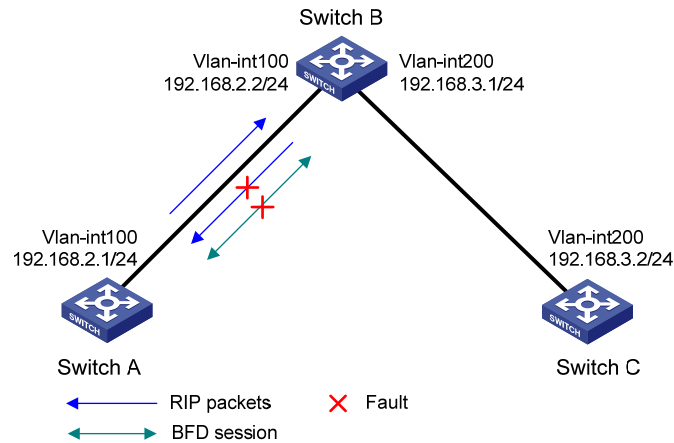
BFD for RIP configuration example (single hop echo detection for a specific destination)

Network requirements

As shown in [Figure 8](#), VLAN-interface 100 of Switch A and Switch B runs RIP process 1. VLAN-interface 200 of Switch B and Switch C runs RIP process 1.

- Configure a static route destined for 100.1.1.0/24 and enable static route redistribution into RIP on both Switch A and Switch C. This allows Switch B to learn two routes destined for 100.1.1.0/24 through VLAN-interface 100 and VLAN-interface 200. The route redistributed from Switch A has a smaller cost than that redistributed from Switch C, so Switch B uses the route through VLAN-interface 200.
- Enable BFD for RIP on VLAN-interface 100 of Switch A, and specify VLAN-interface 100 of Switch B as the destination. When a unidirectional link occurs between Switch A and Switch B, BFD can quickly detect the link failure and notify RIP. Switch B then deletes the neighbor relationship and the route information learned on VLAN-interface 100. It does not receive or send any packets from VLAN-interface 100. When the route learned from Switch A ages out, Switch B uses the route destined for 100.1.1.1 24 through VLAN-interface 200.

Figure 8 Network diagram



Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)
2. Configure basic RIP and enable BFD on the interfaces:

Configure Switch A.

```
<SwitchA> system-view
[SwitchA] rip 1
[SwitchA-rip-1] network 192.168.2.0
[SwitchA-rip-1] import-route static
[SwitchA-rip-1] quit
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] rip bfd enable destination 192.168.2.2
[SwitchA-Vlan-interface100] quit
```

Configure Switch B.

```
<SwitchB> system-view
[SwitchB] rip 1
[SwitchB-rip-1] network 192.168.2.0
[SwitchB-rip-1] network 192.168.3.0
[SwitchB-rip-1] quit
```

Configure Switch C.

```
<SwitchC> system-view
[SwitchC] rip 1
[SwitchC-rip-1] network 192.168.3.0
[SwitchC-rip-1] import-route static cost 3
[SwitchC-rip-1] quit
```

3. Configure BFD parameters on VLAN-interface 100 of Switch A.

```
[SwitchA] bfd echo-source-ip 11.11.11.11
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] bfd min-echo-receive-interval 500
[SwitchA-Vlan-interface100] return
```

4. Configure static routes:

Configure a static route on Switch A.

```
[SwitchA] ip route-static 100.1.1.0 24 null 0
```

Configure a static route on Switch C.

```
[SwitchA] ip route-static 100.1.1.0 24 null 0
```

Verifying the configuration

Display BFD session information on Switch A.

```
<SwitchA> display bfd session
```

```
Total Session Num: 1      Up Session Num: 1      Init Mode: Active
```

```
IPv4 session working under Echo mode:
```

LD	SourceAddr	DestAddr	State	Holdtime	Interface
3	192.168.2.1	192.168.2.2	Up	2000ms	vlan100

Display routes destined for 100.1.1.0/24 on Switch B.

```
<SwitchB> display ip routing-table 100.1.1.0 24 verbose
```

```
Summary Count : 1
```

```
Destination: 100.1.1.0/24
```

```
Protocol: RIP                Process ID: 1
SubProtID: 0x1                Age: 00h02m47s
Cost: 1                       Preference: 100
Tag: 0                        State: Active Adv
OrigTblID: 0x0                OrigVrf: default-vrf
TableID: 0x2                  OrigAs: 0
NBRID: 0x12000002            LastAs: 0
AttrID: 0xffffffff           Neighbor: 192.168.2.1
Flags: 0x1008c               OrigNextHop: 192.168.2.1
Label: NULL                   RealNextHop: 192.168.2.1
BkLabel: NULL                 BkNextHop: N/A
Tunnel ID: Invalid           Interface: vlan-interface 100
BkTunnel ID: Invalid         BkInterface: N/A
```

Display routes destined for 100.1.1.0/24 on Switch B when the link between Switch A and Switch B fails.

```
<SwitchB> display ip routing-table 100.1.1.0 24 verbose
```

```
Summary Count : 1
```

```
Destination: 100.1.1.0/24
```

```
Protocol: RIP                Process ID: 1
SubProtID: 0x1                Age: 00h21m23s
Cost: 4                       Preference: 100
Tag: 0                        State: Active Adv
OrigTblID: 0x0                OrigVrf: default-vrf
TableID: 0x2                  OrigAs: 0
NBRID: 0x12000002            LastAs: 0
AttrID: 0xffffffff           Neighbor: 192.168.3.2
Flags: 0x1008c               OrigNextHop: 192.168.3.2
Label: NULL                   RealNextHop: 192.168.3.2
BkLabel: NULL                 BkNextHop: N/A
```

```
Tunnel ID: Invalid      Interface: vlan-interface 200
BkTunnel ID: Invalid   BkInterface: N/A
```

BFD for RIP configuration example (bidirectional detection in BFD control packet mode)

Network requirements

As shown in [Figure 9](#), VLAN-interface 100 of Switch A and VLAN-interface 200 of Switch C run RIP process 1.

VLAN-interface 300 of Switch A runs RIP process 2. VLAN-interface 400 of Switch C, and VLAN-interface 300 and VLAN-interface 400 of Switch D run RIP process 1.

- Configure a static route destined for 100.1.1.0/24 on Switch A.
- Configure a static route destined for 101.1.1.0/24 on Switch C.
- Enable static route redistribution into RIP on Switch A and Switch C. This allows Switch A to learn two routes destined for 100.1.1.0/24 through VLAN-interface 100 and VLAN-interface 300. It uses the route through VLAN-interface 100.
- Enable BFD on VLAN-interface 100 of Switch A and VLAN-interface 200 of Switch C.

When the link over VLAN-interface 100 fails, BFD can quickly detect the link failure and notify RIP. RIP deletes the neighbor relationship and the route information received learned on VLAN-interface 100. It uses the route destined for 100.1.1.0/24 through VLAN-interface 300.

Figure 9 Network diagram

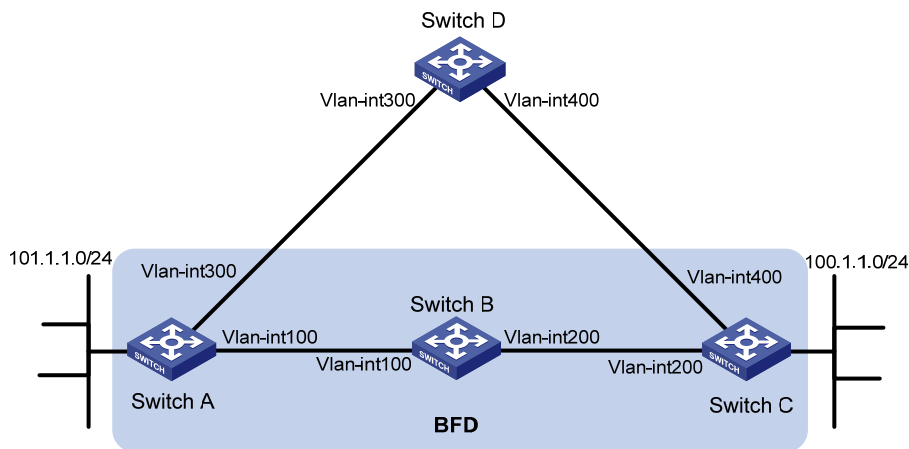


Table 5 Interface and IP address assignment

Device	Interface	IP address
Switch A	VLAN-interface 300	192.168.3.1/24
Switch A	VLAN-interface 100	192.168.1.1/24
Switch B	VLAN-interface 100	192.168.1.2/24
Switch B	VLAN-interface 200	192.168.2.1/24
Switch C	VLAN-interface 200	192.168.2.2/24
Switch C	VLAN-interface 400	192.168.4.2/24
Switch D	VLAN-interface 300	192.168.3.2/24
Switch D	VLAN-interface 400	192.168.4.1/24

Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)
2. Configure basic RIP and enable static route redistribution into RIP so Switch A and Switch C have routes to send to each other:

Configure Switch A.

```
<SwitchA> system-view
[SwitchA] rip 1
[SwitchA-rip-1] version 2
[SwitchA-rip-1] undo summary
[SwitchA-rip-1] network 192.168.1.0
[SwitchA-rip-1] network 101.1.1.0
[SwitchA-rip-1] peer 192.168.2.2
[SwitchA-rip-1] undo validate-source-address
[SwitchA-rip-1] import-route static
[SwitchA-rip-1] quit
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] rip bfd enable
[SwitchA-Vlan-interface100] quit
[SwitchA] rip 2
[SwitchA-rip-2] version 2
[SwitchA-rip-2] undo summary
[SwitchA-rip-2] network 192.168.3.0
[SwitchA-rip-2] quit
```

Configure Switch C.

```
<SwitchC> system-view
[SwitchC] rip 1
[SwitchC-rip-1] version 2
[SwitchC-rip-1] undo summary
[SwitchC-rip-1] network 192.168.2.0
[SwitchC-rip-1] network 192.168.4.0
[SwitchC-rip-1] network 100.1.1.0
[SwitchC-rip-1] peer 192.168.1.1
[SwitchC-rip-1] undo validate-source-address
[SwitchC-rip-1] import-route static
[SwitchC-rip-1] quit
[SwitchC] interface vlan-interface 200
[SwitchC-Vlan-interface200] rip bfd enable
[SwitchC-Vlan-interface200] quit
```

Configure Switch D.

```
<SwitchD> system-view
[SwitchD] rip 1
[SwitchD-rip-1] version 2
[SwitchD-rip-1] undo summary
[SwitchD-rip-1] network 192.168.3.0
[SwitchD-rip-1] network 192.168.4.0
```

3. Configure BFD parameters:

Configure Switch A.

```
[SwitchA] bfd session init-mode active
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] bfd min-transmit-interval 500
[SwitchA-Vlan-interface100] bfd min-receive-interval 500
[SwitchA-Vlan-interface100] bfd detect-multiplier 7
[SwitchA-Vlan-interface100] quit
```

Configure Switch C.

```
[SwitchC] bfd session init-mode active
[SwitchC] interface vlan-interface 200
[SwitchC-Vlan-interface200] bfd min-transmit-interval 500
[SwitchC-Vlan-interface200] bfd min-receive-interval 500
[SwitchC-Vlan-interface200] bfd detect-multiplier 7
[SwitchC-Vlan-interface200] quit
```

4. Configure static routes:

Configure a static route to Switch C on Switch A.

```
[SwitchA] ip route-static 192.168.2.0 24 vlan-interface 100 192.168.1.2
[SwitchA] quit
```

Configure a static route to Switch A on Switch C.

```
[SwitchC] ip route-static 192.168.1.0 24 vlan-interface 200 192.168.2.1
```

Verifying the configuration

Display the BFD session information on Switch A.

```
<SwitchA> display bfd session
```

```
Total Session Num: 1      Up Session Num: 1      Init Mode: Active
```

```
IPv4 session working under Ctrl mode:
```

LD/RD	SourceAddr	DestAddr	State	Holdtime	Interface
513/513	192.168.1.1	192.168.2.2	Up	1700ms	vlan100

Display RIP routes destined for 100.1.1.0/24 on Switch A.

```
<SwitchB> display ip routing-table 100.1.1.0 24 verbose
```

```
Summary Count : 1
```

```
Destination: 100.1.1.0/24
```

```
Protocol: RIP          Process ID: 1
SubProtID: 0x1         Age: 00h02m47s
Cost: 1                Preference: 100
Tag: 0                 State: Active Adv
OrigTblID: 0x0         OrigVrf: default-vrf
TableID: 0x2           OrigAs: 0
NBRID: 0x12000002     LastAs: 0
AttrID: 0xffffffff    Neighbor: 192.168.2.2
Flags: 0x1008c        OrigNextHop: 192.168.2.2
Label: NULL           RealNextHop: 192.168.1.2
BkLabel: NULL         BkNextHop: N/A
Tunnel ID: Invalid    Interface: vlan-interface 100
```

BkTunnel ID: Invalid BkInterface: N/A

Display RIP routes destined for 100.1.1.0/24 on Switch A when the link between Switch B and Switch C fails.

```
<SwitchA> display ip routing-table 100.1.1.0 24 verbose
```

Summary Count : 1

Destination: 100.1.1.0/24

Protocol: RIP	Process ID: 2
SubProtID: 0x1	Age: 00h18m40s
Cost: 2	Preference: 100
Tag: 0	State: Active Adv
OrigTblID: 0x0	OrigVrf: default-vrf
TableID: 0x2	OrigAs: 0
NBRID: 0x12000003	LastAs: 0
AttrID: 0xffffffff	Neighbor: 192.168.3.2
Flags: 0x1008c	OrigNextHop: 192.168.3.2
Label: NULL	RealNextHop: 192.168.3.2
BkLabel: NULL	BkNextHop: N/A
Tunnel ID: Invalid	Interface: vlan-interface 300
BkTunnel ID: Invalid	BkInterface: N/A

Configuring PBR

Overview

Policy-based routing (PBR) uses user-defined policies to route packets. A policy can specify the next hop for packets that match specific criteria such as ACLs.

A device forwards received packets using the following process:

1. The device uses PBR to forward matching packets.
2. If the packets do not match the PBR policy or the PBR-based forwarding fails, the device uses the routing table, excluding the default route, to forward the packets.
3. If the routing table-based forwarding fails, the device uses the default next hop or default output interface defined in PBR to forward packets.
4. If the default next hop or default output interface-based forwarding fails, the device uses the default route to forward packets.

PBR includes local PBR and interface PBR:

- Local PBR guides the forwarding of locally generated packets, such as the ICMP packets generated by using the **ping** command.
- Interface PBR guides the forwarding of packets received on an interface only.

Policy

A policy includes match criteria and actions to be taken on the matching packets. A policy can have one or multiple nodes as follows:

- Each node is identified by a node number. A smaller node number has a higher priority.
- A node contains **if-match** and **apply** clauses. An **if-match** clause specifies a match criterion, and an **apply** clause specifies an action.
- A node has a match mode of **permit** or **deny**.

A policy compares packets with nodes in priority order. If a packet matches the criteria on a node, it is processed by the action on the node. Otherwise, it goes to the next node for a match. If the packet does not match the criteria on any node, it is forwarded according to the routing table.

if-match clause

PBR supports the **if-match acl** clause to set an ACL match criterion. You can specify only one **if-match acl** clause for a node.

apply clause

PBR supports the **apply next-hop** clause to set next hops for packets.

Relationship between the match mode and clauses on the node

Does a packet match all the if-match clauses on the node?	Match mode	
	Permit	Deny
Yes.	<ul style="list-style-type: none">• If the node is configured with an apply clause, PBR executes the apply clause on the node and does not compare the packet with the next node.• If the node is configured with no apply clause, the packet is	The packet is forwarded according to the routing table.

Does a packet match all the if-match clauses on the node?	Match mode	
	Permit	Deny
	forwarded according to the routing table.	
No.	PBR compares the packet with the next node.	PBR compares the packet with the next node.

A node that has no **if-match** clauses matches any packet.

PBR and Track

PBR can work with the Track feature to dynamically adapt the availability status of an **apply** clause to the link status of a tracked next hop:

- When the track entry associated with an object changes to **Negative**, the **apply** clause is invalid.
- When the track entry changes to **Positive** or **NotReady**, the **apply** clause is valid.

For more information about Track-PBR collaboration, see *High Availability Configuration Guide*.

PBR configuration task list

Tasks at a glance
(Required.) Configuring a policy : <ul style="list-style-type: none"> • Creating a node • Configuring match criteria for a node • Configuring actions for a node
(Required.) Configuring PBR : <ul style="list-style-type: none"> • Configuring local PBR • Configuring interface PBR

Configuring a policy

Creating a node

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Create a node for a policy, and enter policy node view.	policy-based-route <i>policy-name</i> [deny permit] node <i>node-number</i>	By default, no policy node is created.

Configuring match criteria for a node

Step	Command	Remarks
1. Enter system view.	system-view	N/A

Step	Command	Remarks
2. Enter policy node view.	policy-based-route <i>policy-name</i> [deny permit] node <i>node-number</i>	N/A
3. Configure an ACL match criterion.	if-match acl <i>acl-number</i> { <i>acl-number</i> name <i>acl-name</i> }	By default, no ACL match criterion is configured.

NOTE:

An ACL match criterion uses the specified ACL to match packets regardless of the **permit** or **deny** action and the time range of the ACL. If the specified ACL does not exist, no packet can match the criterion.

Configuring actions for a node

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter policy node view.	policy-based-route <i>policy-name</i> [deny permit] node <i>node-number</i>	N/A
3. Set next hops.	apply next-hop { <i>ip-address</i> [direct] [track <i>track-entry-number</i>] }&<1- <i>n</i> >	By default, no next hop is specified. You can specify multiple next hops for backup by executing this command once or multiple times. You can specify a maximum of two next hops for a node.

Configuring PBR

Configuring local PBR

Configure PBR by applying a policy locally. PBR uses the policy to guide the forwarding of locally generated packets. The specified policy must already exist. Otherwise, the local PBR configuration fails.

You can apply only one policy locally. Before you apply a new policy, you must first remove the current policy.

Local PBR might affect local services, such as ping and Telnet. Do not configure local PBR unless doing so is required.

To configure local PBR:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Apply a policy locally.	ip local policy-based-route <i>policy-name</i>	By default, no policy is locally applied.

Configuring interface PBR

Configure PBR by applying a policy to an interface. PBR uses the policy to guide the forwarding of packets received on the interface. The specified policy must already exist. Otherwise, the interface PBR configuration fails.

You can apply only one policy to an interface. Before you apply a new policy, you must first remove the current policy from the interface.

You can apply a policy to multiple interfaces.

To configure interface PBR:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type interface-number</i>	N/A
3. Apply a policy to the interface.	ip policy-based-route <i>policy-name</i>	By default, no policy is applied to the interface.

Displaying and maintaining PBR

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display PBR policy information.	display ip policy-based-route [policy <i>policy-name</i>]
Display PBR configuration.	display ip policy-based-route setup
Display local PBR configuration and statistics.	display ip policy-based-route local [slot <i>slot-number</i>]
Display interface PBR configuration and statistics.	display ip policy-based-route interface <i>interface-type interface-number</i> [slot <i>slot-number</i>]
Clear PBR statistics.	reset ip policy-based-route statistics [policy <i>policy-name</i>]

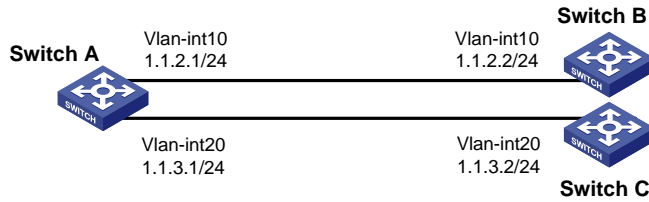
PBR configuration examples

Packet type-based local PBR configuration example

Network requirements

As shown in [Figure 10](#), configure PBR on Switch A to forward all TCP packets to the next hop 1.1.2.2. Switch A forwards other packets according to the routing table.

Figure 10 Network diagram



Configuration procedure

1. Configure Switch A:

Create VLAN 10 and VLAN 20.

```
<SwitchA> system-view
[SwitchA] vlan 10
[SwitchA-vlan10] quit
[SwitchA] vlan 20
[SwitchA-vlan20] quit
```

Configure the IP addresses of VLAN-interface 10 and VLAN-interface 20.

```
[SwitchA] interface vlan-interface 10
[SwitchA-Vlan-interface10] ip address 1.1.2.1 24
[SwitchA-Vlan-interface10] quit
[SwitchA] interface vlan-interface 20
[SwitchA-Vlan-interface20] ip address 1.1.3.1 24
[SwitchA-Vlan-interface20] quit
```

Configure ACL 3101 to match TCP packets.

```
[SwitchA] acl number 3101
[SwitchA-acl-adv-3101] rule permit tcp
[SwitchA-acl-adv-3101] quit
```

Configure Node 5 for policy **aaa** to forward TCP packets to next hop 1.1.2.2.

```
[SwitchA] policy-based-route aaa permit node 5
[SwitchA-pbr-aaa-5] if-match acl 3101
[SwitchA-pbr-aaa-5] apply next-hop 1.1.2.2
[SwitchA-pbr-aaa-5] quit
```

Configure local PBR by applying policy **aaa** to Switch A.

```
[SwitchA] ip local policy-based-route aaa
```

2. Configure Switch B:

Create VLAN 10.

```
<SwitchB> system-view
[SwitchB] vlan 10
[SwitchB-vlan10] quit
```

Configure the IP address of VLAN-interface 10.

```
[SwitchB] interface vlan-interface 10
[SwitchB-Vlan-interface10] ip address 1.1.2.2 24
```

3. Configure Switch C:

Create VLAN 20.

```
<SwitchC> system-view
[SwitchC] vlan 20
[SwitchC-vlan20] quit
```



```
# Configure the IP address of VLAN-interface 20.
[SwitchC] interface vlan-interface 20
[SwitchC-Vlan-interface20] ip address 1.1.3.2 24
```

Verifying the configuration

```
# Telnet to Switch B on Switch A. The operation succeeds.
# Telnet to Switch C on Switch A. The operation fails.
# Ping Switch C from Switch A. The operation succeeds.
Telnet uses TCP and ping uses ICMP. The results show the following:
```

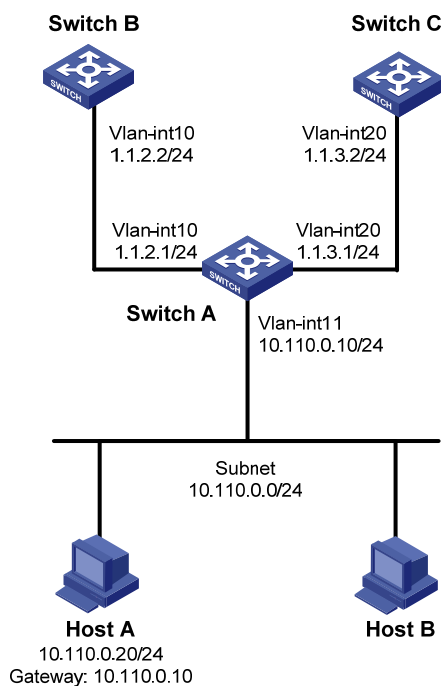
- All TCP packets sent from Switch A are forwarded to the next hop 1.1.2.2.
- Other packets are forwarded through VLAN-interface 20.
- The local PBR configuration is effective.

Packet type-based interface PBR configuration example

Network requirements

As shown in [Figure 11](#), configure PBR on Switch A to forward all TCP packets received on VLAN-interface 11 to the next hop 1.1.2.2. Switch A forwards other packets according to the routing table.

Figure 11 Network diagram



Configuration procedure

1. Configure Switch A:


```
# Create VLAN 10 and VLAN 20.
<SwitchA> system-view
[SwitchA] vlan 10
[SwitchA-vlan10] quit
[SwitchA] vlan 20
```

```
[SwitchA-vlan20] quit
```

Configure the IP addresses of VLAN-interface 10 and VLAN-interface 20.

```
[SwitchA] interface vlan-interface 10
[SwitchA-Vlan-interface10] ip address 1.1.2.1 24
[SwitchA-Vlan-interface10] quit
[SwitchA] interface vlan-interface 20
[SwitchA-Vlan-interface20] ip address 1.1.3.1 24
[SwitchA-Vlan-interface20] quit
```

Configure ACL 3101 to match TCP packets.

```
[SwitchA] acl number 3101
[SwitchA-acl-adv-3101] rule permit tcp
[SwitchA-acl-adv-3101] quit
```

Configure Node 5 for policy **aaa to forward TCP packets to next hop 1.1.2.2.**

```
[SwitchA] policy-based-route aaa permit node 5
[SwitchA-pbr-aaa-5] if-match acl 3101
[SwitchA-pbr-aaa-5] apply next-hop 1.1.2.2
[SwitchA-pbr-aaa-5] quit
```

Configure interface PBR by applying policy **aaa to VLAN-interface 11.**

```
[SwitchA] interface vlan-interface 11
[SwitchA-Vlan-interface11] ip address 10.110.0.10 24
[SwitchA-Vlan-interface11] ip policy-based-route aaa
[SwitchA-Vlan-interface11] quit
```

2. Configure Switch B:

Create VLAN 10.

```
<SwitchB> system-view
[SwitchB] vlan 10
[SwitchB-vlan10] quit
```

Configure the IP address of VLAN-interface 10.

```
[SwitchB] interface vlan-interface 10
[SwitchB-Vlan-interface10] ip address 1.1.2.2 24
[SwitchB-Vlan-interface10] quit
```

Configure a static route to subnet 10.110.0.0/24.

```
[SwitchB] ip route-static 10.110.0.0 24 1.1.2.1
```

3. Configure Switch C:

Create VLAN 20.

```
<SwitchC> system-view
[SwitchC] vlan 20
[SwitchC-vlan20] quit
```

Configure the IP address of VLAN-interface 20.

```
[SwitchC] interface vlan-interface 20
[SwitchC-Vlan-interface20] ip address 1.1.3.2 24
[SwitchC-Vlan-interface20] quit
```

Configure a static route to subnet 10.110.0.0/24.

```
[SwitchC] ip route-static 10.110.0.0 24 1.1.3.1
```

Verifying the configuration

Configure the IP address 10.110.0.20/24 for Host A, and specify its gateway address as 10.110.0.10.

On Host A, Telnet to Switch B that is directly connected to Switch A. The operation succeeds.

On Host A, Telnet to Switch C that is directly connected to Switch A. The operation fails.

Ping Switch C from Host A. The operation succeeds.

Telnet uses TCP and ping uses ICMP. The results show the following:

- All TCP packets arriving on VLAN-interface 11 of Switch A are forwarded to next hop 1.1.2.2.
- Other packets are forwarded through VLAN-interface 20.
- The interface PBR configuration is effective.

Configuring IPv6 static routing

Static routes are manually configured and cannot adapt to network topology changes. If a fault or a topological change occurs in the network, the network administrator must modify the static routes manually. IPv6 static routing works well in a simple IPv6 network.

Configuring an IPv6 static route

Before you configure an IPv6 static route, complete the following tasks:

- Configure parameters for the related interfaces.
- Configure link layer attributes for the related interfaces.
- Make sure the neighboring nodes can reach each other.

To configure an IPv6 static route:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure an IPv6 static route.	ipv6 route-static <i>ipv6-address prefix-length</i> { <i>interface-type interface-number</i> [<i>next-hop-address</i>] <i>next-hop-address</i> } [permanent] [preference <i>preference-value</i>] [tag <i>tag-value</i>] [description <i>description-text</i>]	By default, no IPv6 static route is configured.
3. (Optional.) Configure the default preference for IPv6 static routes.	ipv6 route-static default-preference <i>default-preference-value</i>	The default setting is 60.
4. (Optional.) Delete all IPv6 static routes, including the default route.	delete ipv6 static-routes all	The undo ipv6 route-static command deletes one IPv6 static route.

Configuring BFD for IPv6 static routes

BFD provides a general purpose, standard, and medium- and protocol-independent fast failure detection mechanism. It can uniformly and quickly detect the failures of the bidirectional forwarding paths between two routers for protocols, such as routing protocols. For more information about BFD, see *High Availability Configuration Guide*.

ⓘ IMPORTANT:

Enabling BFD for a flapping route could worsen the situation.

Bidirectional control mode

To use BFD bidirectional control detection between two devices, enable BFD control mode for each device's static route destined to the peer.

To configure a static route and enable BFD control mode, use one of the following methods:

- Specify an output interface and a direct next hop.
- Specify an indirect next hop and a specific BFD packet source address for the static route.

To configure BFD control mode for an IPv6 static route (direct next hop):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure BFD control mode for an IPv6 static route.	ipv6 route-static <i>ipv6-address prefix-length interface-type interface-number next-hop-address</i> bfd control-packet [preference <i>preference-value</i>] [tag <i>tag-value</i>] [description <i>description-text</i>]	By default, BFD control mode for an IPv6 static route is not configured.

To configure BFD control mode for an IPv6 static route (indirect next hop):

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure BFD control mode for an IPv6 static route.	ipv6 route-static <i>ipv6-address prefix-length { next-hop-address</i> bfd control-packet bfd-source <i>ipv6-address</i> } [preference <i>preference-value</i>] [tag <i>tag-value</i>] [description <i>description-text</i>]	By default, BFD control mode for an IPv6 static route is not configured.

Single-hop echo mode

With BFD echo mode enabled for a static route, the output interface sends BFD echo packets to the destination device, which loops the packets back to test the link reachability.



IMPORTANT:

Do not use BFD for a static route with the output interface in spoofing state.

To configure BFD echo mode for an IPv6 static route:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure the source address of echo packets.	bfd echo-source-ipv6 <i>ipv6-address</i>	By default, the source address of echo packets is not configured. The source address of echo packets must be a global unicast address. For more information about this command, see <i>High Availability Command Reference</i> .

Step	Command	Remarks
3. Configure BFD echo mode for an IPv6 static route.	ipv6 route-static <i>ipv6-address prefix-length interface-type interface-number next-hop-address</i> bfd echo-packet [preference <i>preference-value</i>] [tag <i>tag-value</i>] [description <i>description-text</i>]	By default, BFD echo mode for an IPv6 static route is not configured. The next hop IPv6 address must be a global unicast address.

Displaying and maintaining IPv6 static routes

Execute **display** commands in any view.

Task	Command
Display IPv6 static route information.	display ipv6 routing-table protocol static [inactive verbose]
Display IPv6 static route next hop information.	display ipv6 route-static nib [<i>nib-id</i>] [verbose]
Display IPv6 static routing table information.	display ipv6 route-static routing-table [<i>ipv6-address prefix-length</i>]

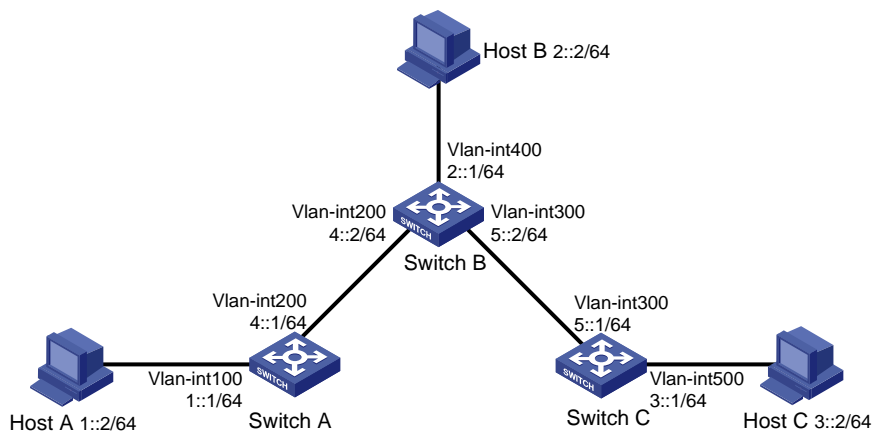
IPv6 static routing configuration examples

Basic IPv6 static route configuration example

Network requirements

As shown in [Figure 12](#), configure IPv6 static routes so that hosts can reach one another.

Figure 12 Network diagram



Configuration procedure

1. Configure the IPv6 addresses for all VLAN interfaces. (Details not shown.)

2. Configure IPv6 static routes:

Configure a default IPv6 static route on Switch A.

```
<SwitchA> system-view
[SwitchA] ipv6 route-static :: 0 4::2
```

Configure two IPv6 static routes on Switch B.

```
<SwitchB> system-view
[SwitchB] ipv6 route-static 1:: 64 4::1
[SwitchB] ipv6 route-static 3:: 64 5::1
```

Configure a default IPv6 static route on Switch C.

```
<SwitchC> system-view
[SwitchC] ipv6 route-static :: 0 5::2
```

3. Configure the IPv6 addresses for all the hosts and configure the default gateway of Host A, Host B, and Host C as 1::1, 2::1, and 3::1.

Verifying the configuration

Display the IPv6 static route information on Switch A.

```
[SwitchA] display ipv6 routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

```
Destination: ::                               Protocol : Static
NextHop      : 4::2                             Preference: 60
Interface    : Vlan-interface200                Cost      : 0
```

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

Display the IPv6 static route information on Switch B.

```
[SwitchB] display ipv6 routing-table protocol static
```

```
Summary Count : 2
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 2
```

```
Destination: 1::/64                           Protocol : Static
NextHop      : 4::1                             Preference: 60
Interface    : Vlan-interface200                Cost      : 0
```

```
Destination: 3::/64                           Protocol : Static
NextHop      : 5::1                             Preference: 60
Interface    : Vlan-interface300                Cost      : 0
```

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

Use the **ping** command to test the reachability.

```

[SwitchA] ping ipv6 3::1
Ping6(104=40+8+56 bytes) 4::1 --> 3::1, press CTRL_C to break
56 bytes from 3::1, icmp_seq=0 hlim=62 time=0.700 ms
56 bytes from 3::1, icmp_seq=1 hlim=62 time=0.351 ms
56 bytes from 3::1, icmp_seq=2 hlim=62 time=0.338 ms
56 bytes from 3::1, icmp_seq=3 hlim=62 time=0.373 ms
56 bytes from 3::1, icmp_seq=4 hlim=62 time=0.316 ms

--- Ping6 statistics for 3::1 ---
5 packet(s) transmitted, 5 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.316/0.416/0.700/0.143 ms

```

BFD for IPv6 static routes configuration example (direct next hop)

Network requirements

As shown in [Figure 13](#):

- Configure an IPv6 static route to subnet 120::/64 on Switch A.
- Configure an IPv6 static route to subnet 121::/64 on Switch B.
- Enable BFD for both routes.
- Configure an IPv6 static route to subnet 120::/64 and an IPv6 static route to subnet 121::/64 on Switch C.

When the link between Switch A and Switch B through the Layer 2 switch fails, BFD can detect the failure immediately, and Switch A and Switch B can communicate through Switch C.

Figure 13 Network diagram

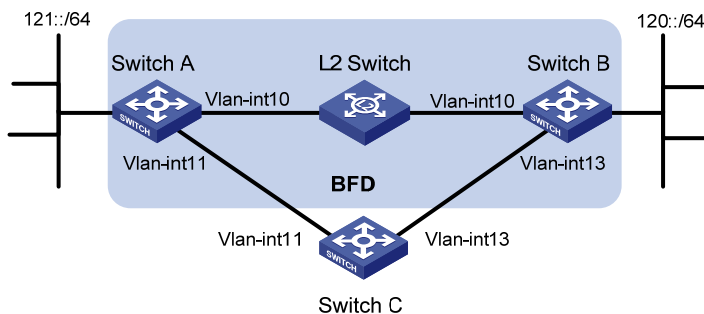


Table 6 Interface and IP address assignment

Device	Interface	IPv6 address	Device	Interface	IPv6 address
Switch A	Vlan-int10	12::1/64	Switch B	Vlan-int13	13::1/64
Switch A	Vlan-int11	10::102/64	Switch C	Vlan-int11	10:: 100/64
Switch B	Vlan-int10	12::2/64	Switch C	Vlan-int13	13::2/64

Configuration procedure

1. Configure IPv6 addresses for interfaces. (Details not shown.)
2. Configure IPv6 static routes and BFD:

Configure IPv6 static routes on Switch A and enable BFD control mode for the static route that traverses the Layer 2 switch.

```
<SwitchA> system-view
[SwitchA] interface vlan-interface 10
[SwitchA-vlan-interface10] bfd min-transmit-interval 500
[SwitchA-vlan-interface10] bfd min-receive-interval 500
[SwitchA-vlan-interface10] bfd detect-multiplier 9
[SwitchA-vlan-interface10] quit
[SwitchA] ipv6 route-static 120:: 64 vlan-interface 10 FE80::2E0:FCFF:FE58:123E bfd
control-packet
[SwitchA] ipv6 route-static 120:: 64 10::100 preference 65
[SwitchA] quit
```

Configure IPv6 static routes on Switch B and enable BFD control mode for the static route that traverses the Layer 2 switch.

```
<SwitchB> system-view
[SwitchB] interface vlan-interface 10
[SwitchB-vlan-interface10] bfd min-transmit-interval 500
[SwitchB-vlan-interface10] bfd min-receive-interval 500
[SwitchB-vlan-interface10] bfd detect-multiplier 9
[SwitchB-vlan-interface10] quit
[SwitchB] ipv6 route-static 121:: 64 vlan-interface 10 FE80::2A0:FCFF:FE00:580A bfd
control-packet
[SwitchB] ipv6 route-static 121:: 64 vlan-interface 13 13::2 preference 65
[SwitchB] quit
```

Configure IPv6 static routes on Switch C.

```
<SwitchC> system-view
[SwitchC] ipv6 route-static 120:: 64 13::1
[SwitchC] ipv6 route-static 121:: 64 10::102
```

Verifying the configuration

Display the BFD sessions on Switch A.

```
<SwitchA> display bfd session
```

```
Total Session Num: 1      Up Session Num: 1      Init Mode: Active
```

```
IPv6 Session Working Under Ctrl Mode:
```

```
Local Discr: 513          Remote Discr: 33
Source IP: FE80::2A0:FCFF:FE00:580A (link-local address of VLAN-interface 10 on
Switch A)
Destination IP: FE80::2E0:FCFF:FE58:123E (link-local address of VLAN-interface 10 on
Switch B)
Session State: Up          Interface: Vlan10
Hold Time: 2012ms
```

The output shows that the BFD session has been created.

Display IPv6 static routes on Switch A.

```
<SwitchA> display ipv6 routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
Summary Count : 1
```

Destination:	120::/64	Protocol	: Static
NextHop	: 12::2	Preference:	60
Interface	: Vlan10	Cost	: 0

```
Direct Routing table Status : <Inactive>
Summary Count : 0
```

The output shows that Switch A communicates with Switch B through VLAN-interface 10. The link over VLAN-interface 10 fails.

Display IPv6 static routes on Switch A again.

```
<SwitchA> display ipv6 routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
Summary Count : 1
```

Destination:	120::/64	Protocol	: Static
NextHop	: 10::100	Preference:	65
Interface	: Vlan11	Cost	: 0

```
Static Routing table Status : < Inactive>
Summary Count : 0
```

The output shows that Switch A communicates with Switch B through VLAN-interface 11.

BFD for IPv6 static routes configuration example (indirect next hop)

Network requirements

As shown in [Figure 14](#):

- Switch A has a route to interface Loopback 1 (2::9/128) on Switch B, and the output interface is VLAN-interface 10.
- Switch B has a route to interface Loopback 1 (1::9/128) on Switch A, and the output interface is VLAN-interface 12.
- Switch D has a route to 1::9/128, and the output interface is VLAN-interface 10. It also has a route to 2::9/128, and the output interface is VLAN-interface 12.

Configure the following:

- Configure an IPv6 static route to subnet 120::/64 on Switch A.
- Configure an IPv6 static route to subnet 121::/64 on Switch B.
- Enable BFD for both routes.
- Configure an IPv6 static route to subnet 120::/64 and an IPv6 static route to subnet 121::/64 on both Switch C and Switch D.

When the link between Switch A and Switch B through Switch D fails, BFD can detect the failure immediately and Switch A and Switch B can communicate through Switch C.

Figure 14 Network diagram

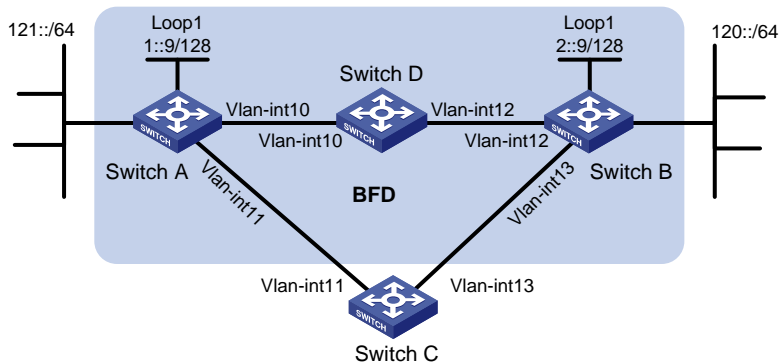


Table 7 Interface and IP address assignment

Device	Interface	IPv6 address	Device	Interface	IPv6 address
Switch A	Vlan-int10	12::1/64	Switch B	Vlan-int12	11::2/64
Switch A	Vlan-int11	10::102/64	Switch B	Vlan-int13	13::1/64
Switch A	Loop1	1::9/128	Switch B	Loop1	2::9/128
Switch C	Vlan-int11	10::100/64	Switch D	Vlan-int10	12::2/64
Switch C	Vlan-int13	13::2/64	Switch D	Vlan-int12	11::1/64

Configuration procedure

1. Configure IPv6 addresses for interfaces. (Details not shown.)
2. Configure IPv6 static routes and BFD:
 - # Configure IPv6 static routes on Switch A and enable BFD control packet mode for the IPv6 static route that traverses Switch D.

```

<SwitchA> system-view
[SwitchA] bfd multi-hop min-transmit-interval 500
[SwitchA] bfd multi-hop min-receive-interval 500
[SwitchA] bfd multi-hop detect-multiplier 9
[SwitchA] ipv6 route-static 120:: 64 2::9 bfd control-packet bfd-source 1::9
[SwitchA] ipv6 route-static 120:: 64 10::100 preference 65
[SwitchA] quit

```

 - # Configure IPv6 static routes on Switch B and enable BFD control packet mode for the static route that traverses Switch D.

```

<SwitchB> system-view
[SwitchB] bfd multi-hop min-transmit-interval 500
[SwitchB] bfd multi-hop min-receive-interval 500
[SwitchB] bfd multi-hop detect-multiplier 9
[SwitchB] ipv6 route-static 121:: 64 1::9 bfd control-packet bfd-source 2::9
[SwitchB] ipv6 route-static 121:: 64 13::2 preference 65
[SwitchB] quit

```

 - # Configure IPv6 static routes on Switch C.

```

<SwitchC> system-view
[SwitchC] ipv6 route-static 120:: 64 13::1

```

```
[SwitchC] ipv6 route-static 121:: 64 10::102
# Configure IPv6 static routes on Switch D.
<SwitchD> system-view
[SwitchD] ipv6 route-static 120:: 64 11::2
[SwitchD] ipv6 route-static 121:: 64 12::1
```

Verifying the configuration

Display the BFD sessions on Switch A.

```
<SwitchA> display bfd session
```

```
Total Session Num: 1      Up Session Num: 1      Init Mode: Active
```

```
IPv6 Session Working Under Ctrl Mode:
```

```
Local Discr: 513          Remote Discr: 33
Source IP: FE80::1:1B49 (link-local address of Loopback1 on Switch A)
Destination IP: FE80::1:1B49 (link-local address of Loopback1 on Switch B)
Session State: Up          Interface: N/A
Hold Time: 2012ms
```

The output shows that the BFD session has been created.

Display the IPv6 static routes on Switch A.

```
<SwitchA> display ipv6 routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

```
Destination: 120::/64          Protocol : Static
NextHop      : 2::9             Preference: 60
Interface    : Vlan10           Cost      : 0
```

```
Static Routing table Status : <Inactive>
```

```
Summary Count : 0
```

The output shows that Switch A communicates Switch B through VLAN-interface 10. The link over VLAN-interface 10 fails.

Display IPv6 static routes on Switch A again.

```
<SwitchA> display ipv6 routing-table protocol static
```

```
Summary Count : 1
```

```
Static Routing table Status : <Active>
```

```
Summary Count : 1
```

```
Destination: 120::/64          Protocol : Static
NextHop      : 10::100          Preference: 65
Interface    : Vlan11           Cost      : 0
```

Static Routing table Status : <Inactive>

Summary Count : 0

The output shows that Switch A communicates with Switch B through VLAN-interface 11.

Configuring an IPv6 default route

A default IPv6 route is used to forward packets that match no entry in the routing table.

A default IPv6 route can be configured in either of the following ways:

- The network administrator can configure a default route with a destination prefix of `::/0`. For more information, see "[Configuring an IPv6 static route.](#)"
- Some dynamic routing protocols, such as RIPng, can generate a default IPv6 route. For example, an upstream router running RIPng can generate a default IPv6 route and advertise it to other routers. These routers install the default IPv6 route with the next hop being the upstream router. For more information, see the respective chapters on those routing protocols in this configuration guide.

Configuring RIPng

Overview

RIP next generation (RIPng) is an extension of RIP-2 for support of IPv6. Most RIP concepts are applicable to RIPng.

RIPng is a distance vector routing protocol. It employs UDP to exchange route information through port 521. RIPng uses a hop count to measure the distance to a destination. The hop count is the metric or cost. The hop count from a router to a directly connected network is 0. The hop count between two directly connected routers is 1. When the hop count is greater than or equal to 16, the destination network or host is unreachable.

By default, the routing update is sent every 30 seconds. If the router receives no routing updates from a neighbor within 180 seconds, the routes learned from the neighbor are considered unreachable. If no routing update is received within another 240 seconds, the router removes these routes from the routing table.

RIPng for IPv6 has the following differences from RIP:

- **UDP port number**—RIPng uses UDP port 521 to send and receive routing information.
- **Multicast address**—RIPng uses FF02::9 as the link-local-router multicast address.
- **Destination Prefix**—128-bit destination address prefix.
- **Next hop**—128-bit IPv6 address.
- **Source address**—RIPng uses FE80::/10 as the link-local source address.

RIPng route entries

RIPng stores route entries in a database. Each route entry contains the following elements:

- **Destination address**—IPv6 address of a destination host or a network.
- **Next hop address**—IPv6 address of the next hop.
- **Egress interface**—Egress interface of the route.
- **Metric**—Cost from the local router to the destination.
- **Route time**—Time elapsed since the most recent update. The time is reset to 0 every time the route entry is updated.
- **Route tag**—Used for route control. For more information, see "Configuring routing policies."

RIPng packets

RIPng uses request and response packets to exchange routing information as follows:

1. When RIPng starts or needs to update some route entries, it sends a multicast request packet to neighbors.
2. When a RIPng neighbor receives the request packet, it sends back a response packet that contains the local routing table. RIPng can also advertise route updates in response packets periodically or advertise a triggered update caused by a route change.
3. After RIPng receives the response, it checks the validity of the response before adding routes to its routing table, including the following details:
 - Whether the source IPv6 address is the link-local address.
 - Whether the port number is correct.

4. A response packet that fails the check is discarded.

Protocols and standards

- RFC 2080, *RIPng for IPv6*
- RFC 2081, *RIPng Protocol Applicability Statement*

RIPng configuration task list

Tasks at a glance
(Required.) Configuring basic RIPng
(Optional.) Configuring RIPng route control : <ul style="list-style-type: none"> • Configuring an additional routing metric • Configuring RIPng route summarization • Advertising a default route • Configuring received/redistributed route filtering • Configuring a preference for RIPng • Configuring RIPng route redistribution
(Optional.) Tuning and optimizing the RIPng network : <ul style="list-style-type: none"> • Configuring RIPng timers • Configuring split horizon and poison reverse • Configuring zero field check on RIPng packets
(Optional.) Configuring RIPng GR
(Optional.) Applying an IPsec profile

Configuring basic RIPng

Before you configure basic RIPng, configure IPv6 addresses for interfaces to ensure IPv6 connectivity between neighboring nodes.

To configure basic RIPng:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Create a RIPng process and enter its view.	ripng [process-id]	By default, the RIPng process is not created.
3. Return to system view.	quit	N/A
4. Enter interface view.	interface interface-type interface-number	N/A
5. Enable RIPng on the interface.	ripng process-id enable	By default, RIPng is disabled. If RIPng is not enabled on an interface, the interface does not send or receive any RIPng route.

Configuring RIPng route control

Before you configure RIPng, complete the following tasks:

- Configure IPv6 addresses for interfaces to ensure IPv6 connectivity between neighboring nodes.
- Configure basic RIPng.

Configuring an additional routing metric

An additional routing metric (hop count) can be added to the metric of an inbound or outbound RIPng route.

An outbound additional metric is added to the metric of a sent route, and it does not change the route's metric in the routing table.

An inbound additional metric is added to the metric of a received route before the route is added into the routing table, and the route's metric is changed.

To configure an inbound or outbound additional routing metric:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Specify an inbound additional routing metric.	ripng metricin <i>value</i>	The default setting is 0.
4. Specify an outbound additional routing metric.	ripng metricout <i>value</i>	The default setting is 1.

Configuring RIPng route summarization

Configure route summarization on an interface, so RIPng advertises a summary route based on the longest match.

RIPng route summarization improves network scalability, reduces routing table size, and increases routing table lookup efficiency.

RIPng advertises a summary route with the smallest metric of all the specific routes.

For example, RIPng has two specific routes to be advertised through an interface: 1:11:11::24 with a metric of a 2 and 1:11:12::34 with a metric of 3. Configure route summarization on the interface, so RIPng advertises a single route 11::0/16 with a metric of 2.

To configure RIPng route summarization:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Advertise a summary IPv6 prefix.	ripng summary-address <i>ipv6-address</i> <i>prefix-length</i>	By default, the summary IPv6 prefix is not configured.

Advertising a default route

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Configure RIPng to advertise a default route.	ripng default-route { only originate } [cost <i>cost</i>]	By default, RIPng does not advertise a default route. This command advertises a default route on the current interface regardless of whether the default route is available in the local IPv6 routing table.

Configuring received/redistributed route filtering

Perform this task to filter received or redistributed routes by using an IPv6 ACL or IPv6 prefix list. You can also configure RIPng to filter routes redistributed from other routing protocols and routes from a specified neighbor.

To configure a RIPng route filtering policy:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIPng view.	ripng [<i>process-id</i>]	N/A
3. Configure a filter policy to filter received routes.	filter-policy { <i>acl6-number</i> prefix-list <i>prefix-list-name</i> } import	By default, RIPng does not filter received routes.
4. Configure a filter policy to filter redistributed routes.	filter-policy { <i>acl6-number</i> prefix-list <i>prefix-list-name</i> } export [<i>protocol</i> [<i>process-id</i>]]	By default, RIPng does not filter redistributed routes.

Configuring a preference for RIPng

Routing protocols each have a preference. When they find routes to the same destination, the route found by the routing protocol with the highest preference is selected as the optimal route. You can manually set a preference for RIPng. The smaller the value, the higher the preference.

To configure a preference for RIPng:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIPng view.	ripng [<i>process-id</i>]	N/A
3. Configure a preference for RIPng.	preference [route-policy <i>route-policy-name</i>] <i>value</i>	The default setting is 100.

Configuring RIPng route redistribution

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIPng view.	ripng [<i>process-id</i>]	N/A
3. Redistribute routes from other routing protocols.	import-route <i>protocol</i> [<i>process-id</i>] [cost <i>cost</i> route-policy <i>route-policy-name</i>] *	By default, RIPng does not redistribute routes from other routing protocols.
4. (Optional.) Configure a default routing metric for redistributed routes.	default cost <i>cost</i>	The default metric of redistributed routes is 0.

Tuning and optimizing the RIPng network

This section describes how to tune and optimize the performance of the RIPng network as well as applications under special network environments.

Before you tune and optimize the RIPng network, complete the following tasks:

- Configure IPv6 addresses for interfaces to ensure IPv6 connectivity between neighboring nodes.
- Configure basic RIPng.

Configuring RIPng timers

You can adjust RIPng timers to optimize the performance of the RIPng network.

When you adjust RIPng timers, consider the network performance, and perform unified configurations on routers running RIPng to avoid unnecessary network traffic or route oscillation.

To configure RIPng timers:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIPng view.	ripng [<i>process-id</i>]	N/A
3. Configure RIPng timers.	timers { garbage-collect <i>garbage-collect-value</i> suppress <i>suppress-value</i> timeout <i>timeout-value</i> update <i>update-value</i> } *	By default: <ul style="list-style-type: none"> • The update timer is 30 seconds. • The timeout timer is 180 seconds. • The suppress timer is 120 seconds. • The garbage-collect timer is 120 seconds.

Configuring split horizon and poison reverse

If both split horizon and poison reverse are configured, only the poison reverse function takes effect.

Configuring split horizon

Split horizon disables RIPng from sending routes through the interface where the routes were learned to prevent routing loops between neighbors.

As a best practice, enable split horizon to prevent routing loops in normal cases.

To configure split horizon:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Enable split horizon.	ripng split-horizon	By default, split horizon is enabled.

Configuring poison reverse

Poison reverse enables a route learned from an interface to be advertised through the interface. However, the metric of the route is set to 16, which means the route is unreachable.

To configure poison reverse:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Enable poison reverse.	ripng poison-reverse	By default, poison reverse is disabled.

Configuring zero field check on RIPng packets

Some fields in the RIPng packet header must be zero. These fields are called zero fields. You can enable zero field check on incoming RIPng packets. If a zero field of a packet contains a non-zero value, RIPng does not process the packets. If you are certain that all packets are trustworthy, disable the zero field check to save CPU resources.

To configure RIPng zero field check:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIPng view.	ripng [<i>process-id</i>]	N/A
3. Enable the zero field check on incoming RIPng packets.	checkzero	By default, this feature is enabled.

Configuring RIPng GR

GR ensures forwarding continuity when a routing protocol restarts or an active/standby switchover occurs.

Two routers are required to complete a GR process. The following are router roles in a GR process:

- **GR restarter**—Graceful restarting router. It must have GR capability.
- **GR helper**—A neighbor of the GR restarter. It helps the GR restarter to complete the GR process.

After RIPng restarts on a router, the router must learn RIPng routes again and updates its FIB table, which causes network disconnections and route reconvergence.

With the GR feature, the restarting router (known as the "GR restarter") can notify the event to its GR capable neighbors. GR capable neighbors (known as "GR helpers") maintain their adjacencies with the router within a configurable GR interval. During this process, the FIB table of the router does not change. After the restart, the router contacts its neighbors to retrieve its FIB.

By default, a RIPng-enabled device acts as the GR helper. Perform this task on the GR restarter.

To configure GR on the GR restarter:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enable RIPng and enter RIPng view.	ripng [<i>process-id</i>]	N/A
3. Enable the GR capability for RIPng.	graceful-restart	By default, RIPng GR is disabled.

Applying an IPsec profile

To protect routing information and prevent attacks, RIPng supports using an IPsec profile to authenticate protocol packets. For more information about IPsec profiles, see *Security Configuration Guide*.

Outbound RIPng packets carry the Security Parameter Index (SPI) defined in the relevant IPsec profile. A device uses the SPI carried in a received packet to match the configured IPsec profile. If they match, the device accepts the packet. If they do not match, the device discards the packet and does not establish a neighbor relationship with the sending device.

You can configure an IPsec profile for a RIPng process or interface. The IPsec profile configured for a process applies to all packets in the process. The IPsec profile configured for an interface applies to packets on the interface. If an interface and its process each have an IPsec profile configured, the interface uses its own IPsec profile.

To apply an IPsec profile to a process:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter RIPng view.	ripng [<i>process-id</i>]	N/A
3. Apply an IPsec profile to the process.	enable ipsec-profile <i>profile-name</i>	By default, no IPsec profile is applied.

To apply an IPsec profile to an interface:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type</i> <i>interface-number</i>	N/A
3. Apply an IPsec profile to the interface.	ripng ipsec-profile <i>profile-name</i>	By default, no IPsec profile is applied.

Displaying and maintaining RIPng

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display configuration information of a RIPng process.	display ripng [<i>process-id</i>]
Display routes in the RIPng database.	display ripng <i>process-id</i> database [<i>ipv6-address</i> <i>prefix-length</i>]
Display the routing information of a specified RIPng process.	display ripng <i>process-id</i> route [<i>ipv6-address</i> <i>prefix-length</i> [verbose] peer <i>ipv6-address</i> statistics]
Display RIPng interface information.	display ripng <i>process-id</i> interface [<i>interface-type</i> <i>interface-number</i>]
Reset a RIPng process.	reset ripng <i>process-id</i> process
Clear statistics for a RIPng process.	reset ripng <i>process-id</i> statistics

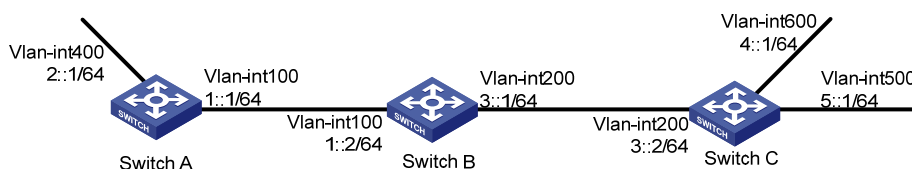
RIPng configuration examples

Basic RIPng configuration example

Network requirements

As shown in Figure 15, all switches run RIPng. Configure Switch B to filter the route 2::/64 learned from Switch A and to forward only the route 4::/64 to Switch A.

Figure 15 Network diagram



Configuration procedure

1. Configure IPv6 addresses for interfaces. (Details not shown.)
2. Configure basic RIPng:

Configure Switch A.

```
<SwitchA> system-view
[SwitchA] ripng 1
[SwitchA-ripng-1] quit
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ripng 1 enable
[SwitchA-Vlan-interface100] quit
[SwitchA] interface vlan-interface 400
[SwitchA-Vlan-interface400] ripng 1 enable
[SwitchA-Vlan-interface400] quit
```

Configure Switch B.

```
<SwitchA> system-view
[SwitchA] ripng 1
[SwitchA-ripng-1] quit
[SwitchA] interface vlan-interface 100
```

```
[SwitchA-Vlan-interface100] ripng 1 enable
[SwitchA-Vlan-interface100] quit
[SwitchA] interface vlan-interface 400
[SwitchA-Vlan-interface400] ripng 1 enable
[SwitchA-Vlan-interface400] quit
```

Configure Switch C.

```
<SwitchC> system-view
[SwitchC] ripng 1
[SwitchC-ripng-1] quit
[SwitchC] interface vlan-interface 200
[SwitchC-Vlan-interface200] ripng 1 enable
[SwitchC-Vlan-interface200] quit
[SwitchC] interface vlan-interface 500
[SwitchC-Vlan-interface500] ripng 1 enable
[SwitchC-Vlan-interface500] quit
[SwitchC] interface vlan-interface 600
[SwitchC-Vlan-interface600] ripng 1 enable
[SwitchC-Vlan-interface600] quit
```

Display the RIPng routing table on Switch B.

```
[SwitchB] display ripng 1 route
Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
              O - Optimal, F - Flush to RIB
```

```
Peer FE80::20F:E2FF:FE23:82F5 on Vlan-interface100
Destination 1::/64,
    via FE80::20F:E2FF:FE23:82F5, cost 1, tag 0, AOF, 6 secs
Destination 2::/64,
    via FE80::20F:E2FF:FE23:82F5, cost 1, tag 0, AOF, 6 secs
```

```
Peer FE80::20F:E2FF:FE00:100 on Vlan-interface200
Destination 3::/64,
    via FE80::20F:E2FF:FE00:100, cost 1, tag 0, AOF, 11 secs
Destination 4::/64,
    via FE80::20F:E2FF:FE00:100, cost 1, tag 0, AOF, 11 secs
Destination 5::/64,
    via FE80::20F:E2FF:FE00:100, cost 1, tag 0, AOF, 11 secs
```

Display the RIPng routing table on Switch A.

```
[SwitchA] display ripng 1 route
Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
              O - Optimal, F - Flush to RIB
```

```
Peer FE80::200:2FF:FE64:8904 on Vlan-interface100
Destination 1::/64,
    via FE80::200:2FF:FE64:8904, cost 1, tag 0, AOF, 31 secs
Destination 3::/64,
    via FE80::200:2FF:FE64:8904, cost 1, tag 0, AOF, 31 secs
```

```

Destination 4::/64,
    via FE80::200:2FF:FE64:8904, cost 2, tag 0, AOF, 31 secs
Destination 5::/64,
    via FE80::200:2FF:FE64:8904, cost 2, tag 0, AOF, 31 secs

```

3. Configure route filtering:

Use IPv6 prefix lists on Switch B to filter received and redistributed routes.

```

[SwitchB] ipv6 prefix-list aaa permit 4:: 64
[SwitchB] ipv6 prefix-list bbb deny 2:: 64
[SwitchB] ipv6 prefix-list bbb permit :: 0 less-equal 128
[SwitchB] ripng 1
[SwitchB-ripng-1] filter-policy prefix-list aaa export
[SwitchB-ripng-1] filter-policy prefix-list bbb import
[SwitchB-ripng-1] quit

```

Display RIPng routing tables on Switch B and Switch A.

```

[SwitchB] display ripng 1 route
    Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
                 O - Optimal, F - Flush to RIB

```

```

-----
Peer FE80::1:100 on Vlan-interface100
Destination 1::/64,
    via FE80::2:100, cost 1, tag 0, AOF, 6 secs

```

```

Peer FE80::3:200 on Vlan-interface200
Destination 3::/64,
    via FE80::2:200, cost 1, tag 0, AOF, 11 secs
Destination 4::/64,
    via FE80::2:200, cost 1, tag 0, AOF, 11 secs
Destination 5::/64,
    via FE80::2:200, cost 1, tag 0, AOF, 11 secs

```

```

[SwitchA] display ripng 1 route
    Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
                 O - Optimal, F - Flush to RIB

```

```

-----
Peer FE80::2:100 on Vlan-interface100
Destination 4::/64,
    via FE80::1:100, cost 2, tag 0, AOF, 2 secs

```

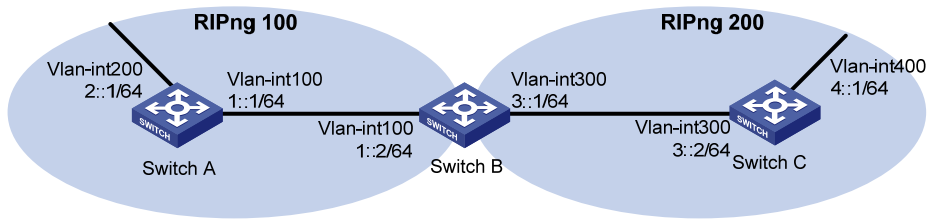
RIPng route redistribution configuration example

Network requirements

As shown in [Figure 16](#), Switch B communicates with Switch A through RIPng 100 and with Switch C through RIPng 200.

Configure route redistribution on Switch B, so the two RIPng processes can redistribute routes from each other.

Figure 16 Network diagram



Configuration procedure

1. Configure IPv6 addresses for interfaces. (Details not shown.)
2. Configure basic RIPng:

Enable RIPng 100 on Switch A.

```
<SwitchA> system-view
[SwitchA] ripng 100
[SwitchA-ripng-100] quit
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ripng 100 enable
[SwitchA-Vlan-interface100] quit
[SwitchA] interface vlan-interface 200
[SwitchA-Vlan-interface200] ripng 100 enable
[SwitchA-Vlan-interface200] quit
```

Enable RIPng 100 and RIPng 200 on Switch B.

```
<SwitchB> system-view
[SwitchB] ripng 100
[SwitchB-ripng-100] quit
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ripng 100 enable
[SwitchB-Vlan-interface100] quit
[SwitchB] ripng 200
[SwitchB-ripng-200] quit
[SwitchB] interface vlan-interface 300
[SwitchB-Vlan-interface300] ripng 200 enable
[SwitchB-Vlan-interface300] quit
```

Enable RIPng 200 on Switch C.

```
<SwitchC> system-view
[SwitchC] ripng 200
[SwitchC] interface vlan-interface 300
[SwitchC-Vlan-interface300] ripng 200 enable
[SwitchC-Vlan-interface300] quit
[SwitchC] interface vlan-interface 400
[SwitchC-Vlan-interface400] ripng 200 enable
[SwitchC-Vlan-interface400] quit
```

Display the routing table on Switch A.

```
[SwitchA] display ipv6 routing-table
```

```
Destinations : 7 Routes : 7
```

```

Destination: ::1/128                                Protocol : Direct
NextHop      : ::1                                  Preference: 0
Interface    : InLoop0                              Cost      : 0

Destination: 1::/64                                 Protocol : Direct
NextHop      : 1::1                                  Preference: 0
Interface    : Vlan100                              Cost      : 0

Destination: 1::1/128                               Protocol : Direct
NextHop      : ::1                                  Preference: 0
Interface    : InLoop0                              Cost      : 0

Destination: 2::/64                                 Protocol : Direct
NextHop      : 2::1                                  Preference: 0
Interface    : Vlan200                              Cost      : 0

Destination: 2::1/128                               Protocol : Direct
NextHop      : ::1                                  Preference: 0
Interface    : InLoop0                              Cost      : 0

Destination: FE80::/10                              Protocol : Direct
NextHop      : ::                                   Preference: 0
Interface    : NULL0                                Cost      : 0

Destination: FF00::/8                               Protocol : Direct
NextHop      : ::                                   Preference: 0
Interface    : NULL0                                Cost      : 0

```

3. Configure RIPng route redistribution:

Configure route redistribution between the two RIPng processes on Switch B.

```

[SwitchB] ripng 100
[SwitchB-ripng-100] import-route ripng 200
[SwitchB-ripng-100] quit
[SwitchB] ripng 200
[SwitchB-ripng-200] import-route ripng 100
[SwitchB-ripng-200] quit

```

Display the routing table on Switch A.

```
[SwitchA] display ipv6 routing-table
```

```
Destinations : 8 Routes : 8
```

```

Destination: ::1/128                                Protocol : Direct
NextHop      : ::1                                  Preference: 0
Interface    : InLoop0                              Cost      : 0

Destination: 1::/64                                 Protocol : Direct
NextHop      : 1::1                                  Preference: 0
Interface    : Vlan100                              Cost      : 0

```

```

Destination: 1::1/128                                Protocol : Direct
NextHop      : ::1                                    Preference: 0
Interface    : InLoop0                               Cost      : 0

Destination: 2::/64                                  Protocol : Direct
NextHop      : 2::1                                    Preference: 0
Interface    : Vlan200                               Cost      : 0

Destination: 2::1/128                                Protocol : Direct
NextHop      : ::1                                    Preference: 0
Interface    : InLoop0                               Cost      : 0

Destination: 4::/64                                  Protocol : RIPng
NextHop      : FE80::200:BFF:FE01:1C02              Preference: 100
Interface    : Vlan100                               Cost      : 1

Destination: FE80::/10                               Protocol : Direct
NextHop      : ::                                     Preference: 0
Interface    : NULL0                                 Cost      : 0

Destination: FF00::/8                               Protocol : Direct
NextHop      : ::                                     Preference: 0
Interface    : NULL0                                 Cost      : 0

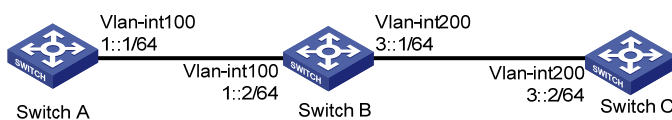
```

RIPng IPsec profile configuration example

Network requirements

As shown in [Figure 17](#), configure RIPng on the switches, and configure IPsec profiles on the switches to authenticate and encrypt protocol packets.

Figure 17 Network diagram



Configuration procedure

1. Configure IPv6 addresses for interfaces. (Details not shown.)
2. Configure RIPng basic functions:

Configure Switch A.

```

<SwitchA> system-view
[SwitchA] ripng 1
[SwitchA-ripng-1] quit
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ripng 1 enable
[SwitchA-Vlan-interface100] quit

```

Configure Switch B.

```

<SwitchB> system-view
[SwitchB] ripng 1

```

```
[SwitchB-ripng-1] quit
[SwitchB] interface vlan-interface 200
[SwitchB-Vlan-interface200] ripng 1 enable
[SwitchB-Vlan-interface200] quit
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ripng 1 enable
[SwitchB-Vlan-interface100] quit
```

Configure Switch C.

```
<SwitchC> system-view
[SwitchC] ripng 1
[SwitchC-ripng-1] quit
[SwitchC] interface vlan-interface 200
[SwitchC-Vlan-interface200] ripng 1 enable
[SwitchC-Vlan-interface200] quit
```

3. Configure RIPng IPsec profiles:

o On Switch A:

Create an IPsec transform set named **protrf1**.

```
[SwitchA] ipsec transform-set protrf1
```

Specify the ESP encryption and authentication algorithms.

```
[SwitchA-ipsec-transform-set-protrf1] esp encryption-algorithm 3des-cbc
```

```
[SwitchA-ipsec-transform-set-protrf1] esp authentication-algorithm md5
```

Specify the encapsulation mode as **transport**.

```
[SwitchA-ipsec-transform-set-protrf1] encapsulation-mode transport
```

```
[SwitchA-ipsec-transform-set-protrf1] quit
```

Create a manual IPsec profile named **profile001**.

```
[SwitchA] ipsec profile profile001 manual
```

Reference IPsec transform set **protrf1**.

```
[SwitchA-ipsec-profile-profile001-manual] transform-set protrf1
```

Configure the inbound and outbound SPIs for ESP.

```
[SwitchA-ipsec-profile-profile001-manual] sa spi inbound esp 256
```

```
[SwitchA-ipsec-profile-profile001-manual] sa spi outbound esp 256
```

Configure the inbound and outbound SA keys for ESP.

```
[SwitchA-ipsec-profile-profile001-manual] sa string-key inbound esp simple abc
```

```
[SwitchA-ipsec-profile-profile001-manual] sa string-key outbound esp simple abc
```

```
[SwitchA-ipsec-profile-profile001-manual] quit
```

o On Switch B:

Create an IPsec transform set named **protrf1**.

```
[SwitchB] ipsec transform-set protrf1
```

Specify the ESP encryption and authentication algorithms.

```
[SwitchB-ipsec-transform-set-protrf1] esp encryption-algorithm 3des-cbc
```

```
[SwitchB-ipsec-transform-set-protrf1] esp authentication-algorithm md5
```

Specify the encapsulation mode as **transport**.

```
[SwitchB-ipsec-transform-set-protrf1] encapsulation-mode transport
```

```
[SwitchB-ipsec-transform-set-protrf1] quit
```

Create a manual IPsec profile named **profile001**.

```
[SwitchB] ipsec profile profile001 manual
```

Reference IPsec transform set **protrf1**.

```
[SwitchB-ipsec-profile-profile001-manual] transform-set protrf1
# Configure the inbound and outbound SPIs for ESP.
[SwitchB-ipsec-profile-profile001-manual] sa spi inbound esp 256
[SwitchB-ipsec-profile-profile001-manual] sa spi outbound esp 256
# Configure the inbound and outbound SA keys for ESP.
[SwitchB-ipsec-profile-profile001-manual] sa string-key inbound esp simple abc
[SwitchB-ipsec-profile-profile001-manual] sa string-key outbound esp simple abc
[SwitchB-ipsec-profile-profile001-manual] quit
```

o On Switch C:

Create an IPsec transform set named **protrf1**.

```
[SwitchC] ipsec transform-set protrf1
```

Specify the ESP encryption and authentication algorithms.

```
[SwitchC-ipsec-transform-set-protrf1] esp encryption-algorithm 3des-cbc
```

```
[SwitchC-ipsec-transform-set-protrf1] esp authentication-algorithm md5
```

Specify the encapsulation mode as **transport**.

```
[SwitchC-ipsec-transform-set-protrf1] encapsulation-mode transport
```

```
[SwitchC-ipsec-transform-set-protrf1] quit
```

Create a manual IPsec profile named **profile001**.

```
[SwitchC] ipsec profile profile001 manual
```

Reference IPsec transform set **protrf1**.

```
[SwitchC-ipsec-profile-profile001-manual] transform-set protrf1
```

Configure the inbound and outbound SPIs for ESP.

```
[SwitchC-ipsec-profile-profile001-manual] sa spi inbound esp 256
```

```
[SwitchC-ipsec-profile-profile001-manual] sa spi outbound esp 256
```

Configure the inbound and outbound SA keys for ESP.

```
[SwitchC-ipsec-profile-profile001-manual] sa string-key inbound esp simple abc
```

```
[SwitchC-ipsec-profile-profile001-manual] sa string-key outbound esp simple abc
```

```
[SwitchC-ipsec-profile-profile001-manual] quit
```

4. Apply the IPsec profiles to the RIPng process:

Configure Switch A.

```
[SwitchA] ripng 1
```

```
[SwitchA-ripng-1] enable ipsec-profile profile001
```

```
[SwitchA-ripng-1] quit
```

Configure Switch B.

```
[SwitchB] ripng 1
```

```
[SwitchB-ripng-1] enable ipsec-profile profile001
```

```
[SwitchB-ripng-1] quit
```

Configure Switch C.

```
[SwitchC] ripng 1
```

```
[SwitchC-ripng-1] enable ipsec-profile profile001
```

```
[SwitchC-ripng-1] quit
```

Verifying the configuration

Verify that RIPng packets between Switches A, B and C are protected by IPsec. (Details not shown.)

Configuring IPv6 PBR

Overview

Policy-based routing (PBR) uses user-defined policies to route packets. A policy can specify the next hop for packets that match specific criteria such as ACLs.

A device forwards received packets using the following process:

1. The device uses PBR to forward matching packets.
2. If the packets do not match the PBR policy or the PBR-based forwarding fails, the device uses the routing table, excluding the default route, to forward the packets.
3. If the routing table-based forwarding fails, the device uses the default next hop or default output interface defined in PBR to forward packets.
4. If the default next hop or default output interface-based forwarding fails, the device uses the default route to forward packets.

PBR includes local PBR and interface PBR:

- Local PBR guides the forwarding of locally generated packets, such as the ICMP packets generated by using the **ping** command.
- Interface PBR guides the forwarding of packets received on an interface only.

Policy

An IPv6 policy includes match criteria and actions to be taken on the matching packets. A policy can have one or multiple nodes as follows:

- Each node is identified by a node number. A smaller node number has a higher priority.
- A node contains **if-match** and **apply** clauses. An **if-match** clause specifies a match criterion, and an **apply** clause specifies an action.
- A node has a match mode of **permit** or **deny**.

An IPv6 policy compares packets with nodes in priority order. If a packet matches the criteria on a node, it is processed by the action on the node. Otherwise, it goes to the next node for a match. If the packet does not match the criteria on any node, it is forwarded according to the routing table.

if-match clause

IPv6 PBR supports the **if-match acl** clause to set an ACL match criterion. You can specify only one **if-match acl** clause for a node.

apply clause

IPv6 PBR supports the **apply next-hop** clause to set next hops for packets.

Relationship between the match mode and clauses on the node

Does a packet match all the if-match clauses on the node?	Match mode	
	In permit mode	In deny mode
Yes	<ul style="list-style-type: none">• If the node is configured with an apply clause, IPv6 PBR executes the apply clause on the node and does not compare the packet with the next node.	The packet is forwarded according to the routing table.

Does a packet match all the if-match clauses on the node?	Match mode	
	In permit mode	In deny mode
	<ul style="list-style-type: none"> If the node is configured with no apply clause, the packet is forwarded according to the routing table. 	
No	IPv6 PBR compares the packet with the next node.	IPv6 PBR compares the packet with the next node.

A node that has no **if-match** clauses matches any packet.

PBR and Track

PBR can work with the Track feature to dynamically adapt the availability status of an **apply** clause to the link status of a tracked next hop.

- When the track entry associated with an object changes to **Negative**, the **apply** clause is invalid.
- When the track entry changes to **Positive** or **NotReady**, the **apply** clause is valid.

For more information about Track-PBR collaboration, see *High Availability Configuration Guide*.

IPv6 PBR configuration task list

Tasks at a glance
(Required.) Configuring an IPv6 policy : <ul style="list-style-type: none"> Creating an IPv6 node Configuring match criteria for an IPv6 node Configuring actions for an IPv6 node
(Required.) Configuring IPv6 PBR : <ul style="list-style-type: none"> Configuring IPv6 local PBR Configuring IPv6 interface PBR

Configuring an IPv6 policy

Creating an IPv6 node

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Create an IPv6 policy or policy node, and enter IPv6 policy node view.	ipv6 policy-based-route <i>policy-name</i> [deny permit] node <i>node-number</i>	By default, no IPv6 policy node is created.

Configuring match criteria for an IPv6 node

Step	Command	Remarks
------	---------	---------

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter IPv6 policy node view.	ipv6 policy-based-route <i>policy-name</i> [deny permit] node <i>node-number</i>	N/A
3. Configure an ACL match criterion.	if-match acl { <i>acl6-number</i> name <i>acl6-name</i> }	By default, no ACL match criterion is configured.

NOTE:

An ACL match criterion uses the specified ACL to match packets regardless of the **permit** or **deny** action and the time range of the ACL. If the specified ACL does not exist, no packet can match the criterion.

Configuring actions for an IPv6 node

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter IPv6 policy node view.	ipv6 policy-based-route <i>policy-name</i> [deny permit] node <i>node-number</i>	N/A
3. Set next hops for permitted IPv6 packets.	apply next-hop { <i>ipv6-address</i> [direct] [track <i>track-entry-number</i>] }&<1- <i>n</i> >	By default, no next hop is specified. You can specify multiple next hops for backup by executing this command once or multiple times. You can specify a maximum of two next hops for a node.

Configuring IPv6 PBR

Configuring IPv6 local PBR

Configure IPv6 PBR by applying a policy locally. IPv6 PBR uses the policy to guide the forwarding of locally generated packets. The specified policy must already exist. Otherwise, the IPv6 local PBR configuration fails.

You can apply only one policy locally. Before you apply a new policy, you must first remove the current policy.

IPv6 local PBR might affect local services, such as ping and Telnet. Do not configure IPv6 local PBR unless doing so is required.

To configure IPv6 local PBR:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Apply a policy locally.	ipv6 local policy-based-route <i>policy-name</i>	By default, no policy is locally applied.

Configuring IPv6 interface PBR

Configure IPv6 PBR by applying an IPv6 policy to an interface. IPv6 PBR uses the policy to guide the forwarding of IPv6 packets received on the interface. The specified policy must already exist. Otherwise, the IPv6 interface PBR configuration fails.

You can apply only one policy to an interface. Before you apply a new policy, you must first remove the current policy from the interface.

You can apply a policy to multiple interfaces.

To configure IPv6 interface PBR:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter interface view.	interface <i>interface-type interface-number</i>	N/A
3. Apply an IPv6 policy to the interface.	ipv6 policy-based-route <i>policy-name</i>	By default, no IPv6 policy is applied to the interface.

Displaying and maintaining IPv6 PBR

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display IPv6 PBR policy information.	display ipv6 policy-based-route [policy <i>policy-name</i>]
Display IPv6 PBR configuration.	display ipv6 policy-based-route setup
Display IPv6 local PBR configuration and statistics.	display ipv6 policy-based-route local [slot <i>slot-number</i>]
Display IPv6 interface PBR configuration and statistics.	display ipv6 policy-based-route interface <i>interface-type interface-number</i> [slot <i>slot-number</i>]
Clear IPv6 PBR statistics.	reset ipv6 policy-based-route statistics [policy <i>policy-name</i>]

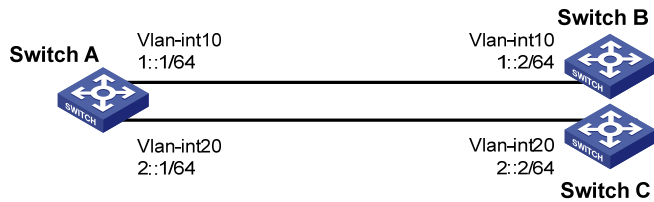
IPv6 PBR configuration examples

Packet type-based IPv6 local PBR configuration example

Network requirements

As shown in [Figure 18](#), configure IPv6 PBR on Switch A to forward all TCP packets to the next hop 1::2. Switch A forwards other packets according to the routing table.

Figure 18 Network diagram



Configuration procedure

1. Configure Switch A:

Create VLAN 10 and VLAN 20.

```
<SwitchA> system-view
[SwitchA] vlan 10
[SwitchA-vlan10] quit
[SwitchA] vlan 20
[SwitchA-vlan20] quit
```

Configure the IPv6 addresses of VLAN-interface 10 and VLAN-interface 20.

```
[SwitchA] interface vlan-interface 10
[SwitchA-Vlan-interface10] ipv6 address 1::1 64
[SwitchA-Vlan-interface10] quit
[SwitchA] interface vlan-interface 20
[SwitchA-Vlan-interface20] ipv6 address 2::1 64
[SwitchA-Vlan-interface20] quit
```

Configure ACL 3001 to match TCP packets.

```
[SwitchA] acl ipv6 number 3001
[SwitchA-acl6-adv-3001] rule permit tcp
[SwitchA-acl6-adv-3001] quit
```

Configure Node 5 for policy **aaa** to forward TCP packets to next hop 1::2.

```
[SwitchA] ipv6 policy-based-route aaa permit node 5
[SwitchA-pbr6-aaa-5] if-match acl 3001
[SwitchA-pbr6-aaa-5] apply next-hop 1::2
[SwitchA-pbr6-aaa-5] quit
```

Configure IPv6 local PBR by applying policy **aaa** to Switch A.

```
[SwitchA] ipv6 local policy-based-route aaa
```

2. Configure Switch B:

Create VLAN 10.

```
<SwitchB> system-view
[SwitchB] vlan 10
[SwitchB-vlan10] quit
```

Configure the IPv6 address of VLAN-interface 10.

```
[SwitchB] interface vlan-interface 10
[SwitchB-Vlan-interface10] ipv6 address 1::2 64
```

3. Configure Switch C:

Create VLAN 20.

```
<SwitchC> system-view
[SwitchC] vlan 20
[SwitchC-vlan20] quit
```

```

# Configure the IPv6 address of VLAN-interface 20.
[SwitchC] interface vlan-interface 20
[SwitchC-Vlan-interface20] ipv6 address 2::2 64

```

Verifying the configuration

```

# Telnet to Switch B on Switch A. The operation succeeds.
# Telnet to Switch C on Switch A. The operation fails.
# Ping Switch C from Switch A. The operation succeeds.

```

Telnet uses TCP, and ping uses ICMP. The results show the following:

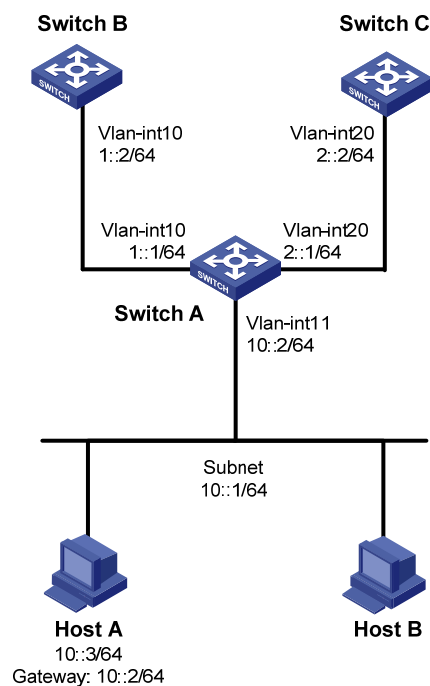
- All TCP packets sent from Switch A are forwarded to the next hop 1::2.
- Other packets are forwarded through VLAN-interface 20.
- The IPv6 local PBR configuration is effective.

Packet type-based IPv6 interface PBR configuration example

Network requirements

As shown in [Figure 19](#), configure IPv6 PBR on Switch A to forward all TCP packets received on VLAN-interface 11 to the next hop 1::2. Switch A forwards other IPv6 packets according to the routing table.

Figure 19 Network diagram



Configuration procedure

1. Configure Switch A:


```

# Create VLAN 10 and VLAN 20.
<SwitchA> system-view
[SwitchA] vlan 10
[SwitchA-vlan10] quit
[SwitchA] vlan 20

```

```
[SwitchA-vlan20] quit
# Configure the IPv6 addresses of VLAN-interface 10 and VLAN-interface 20.
[SwitchA] interface vlan-interface 10
[SwitchA-Vlan-interface10] ipv6 address 1::1 64
[SwitchA-Vlan-interface10] quit
[SwitchA] interface vlan-interface 20
[SwitchA-Vlan-interface20] ipv6 address 1::2 64
[SwitchA-Vlan-interface20] quit
# Configure ACL 3001 to match TCP packets.
[SwitchA] acl ipv6 number 3001
[SwitchA-acl6-adv-3001] rule permit tcp
[SwitchA-acl6-adv-3001] quit
# Configure Node 5 for policy aaa to forward TCP packets to next hop 1::2.
[SwitchA] ipv6 policy-based-route aaa permit node 5
[SwitchA-pbr6-aaa-5] if-match acl 3001
[SwitchA-pbr6-aaa-5] apply next-hop 1::2
[SwitchA-pbr6-aaa-5] quit
# Configure IPv6 interface PBR by applying policy aaa to VLAN-interface 11.
[SwitchA] interface vlan-interface 11
[SwitchA-Vlan-interface11] ipv6 address 10::2 64
[SwitchA-Vlan-interface11] undo ipv6 nd ra halt
[SwitchA-Vlan-interface11] ipv6 policy-based-route aaa
```

2. Configure Switch B:

Create VLAN 10.

```
<SwitchB> system-view
[SwitchB] vlan 10
[SwitchB-vlan10] quit
```

Configure the IPv6 address of VLAN-interface 10.

```
[SwitchB] interface vlan-interface 10
[SwitchB-Vlan-interface10] ipv6 address 1::2 64
[SwitchB-Vlan-interface10] quit
```

Configure an IPv6 static route to subnet 10::1/64.

```
[SwitchB] ipv6 route-static 10::1 64 1::1
```

3. Configure Switch C:

Create VLAN 20.

```
<SwitchC> system-view
[SwitchC] vlan 20
[SwitchC-vlan20] quit
```

Configure the IPv6 address of VLAN-interface 20.

```
[SwitchC] interface vlan-interface 20
[SwitchC-Vlan-interface20] ipv6 address 2::2 64
[SwitchC-Vlan-interface20] quit
```

Configure an IPv6 static route to subnet 10::1/64.

```
[SwitchC] ipv6 route-static 10::1 64 2::1
```

Verifying the configuration

Enable IPv6 and configure the IPv6 address 10::3 for Host A.

```
C:\>ipv6 install
```

Installing...

Succeeded.

```
C:\>ipv6 add 4/10::3
```

On Host A, Telnet to Switch B that is directly connected to Switch A. The operation succeeds.

On Host A, Telnet to Switch C that is directly connected to Switch A. The operation fails.

Ping Switch C from Host A. The operation succeeds.

Telnet uses TCP, and ping uses ICMP. The results show the following:

- All TCP packets arriving on VLAN-interface 11 of Switch A are forwarded to next hop 1::2.
- Other packets are forwarded through VLAN-interface 20.
- The IPv6 interface PBR configuration is effective.

Configuring routing policies

Overview

Routing policies control routing paths by filtering and modifying routing information. This chapter describes both IPv4 and IPv6 routing policies.

Routing policies can filter advertised, received, and redistributed routes, and modify attributes for specific routes.

To configure a routing policy:

1. Configure filters based on route attributes, such as destination address and the advertising router's address.
2. Create a routing policy and apply filters to the routing policy.

Filters

Routing policies can use the following filters to match routes.

ACL

ACLs include IPv4 ACLs and IPv6 ACLs. An ACL can match the destination or next hop of routes.

For more information about ACLs, see *ACL and QoS Configuration Guide*.

IP prefix list

IP prefix lists include IPv4 prefix lists and IPv6 prefix lists.

An IP prefix list matches the destination address of routes. You can use the **gateway** option to receive routes only from specific routers. For more information about the **gateway** option, see "Configuring RIP".

An IP prefix list, identified by name, can contain multiple items. Each item, identified by an index number, specifies a prefix range to match. An item with a smaller index number is matched first. A route that matches one item matches the IP prefix list.

Routing policy

A routing policy can contain multiple nodes, which are in a logical OR relationship. A node with a smaller number is matched first. A route (except the route configured with the **continue** clauses) that matches one node matches the routing policy.

Each node has a match mode of **permit** or **deny**.

- **permit**—Specifies the **permit** match mode for a routing policy node. If a route matches all the **if-match** clauses of the node, it is handled by the **apply** clauses of the node. The route does not compare with the next node unless the **continue** clause is configured. If a route does not match all the **if-match** clauses of the node, it compares with the next node.
- **deny**—Specifies the **deny** match mode for a routing policy node. The **apply** and **continue** clauses of a deny-mode node are never executed. If a route matches all the **if-match** clauses of the node, it is discarded and does not compare with the next node. If a route does not match all the **if-match** clauses of the node, it compares with the next node.

A node can contain a set of **if-match**, **apply**, and **continue** clauses.

- **if-match** clauses—Configure the match criteria that match the attributes of routes. The **if-match** clauses are in a logical AND relationship. A route must match all the **if-match** clauses to match the node.

- **apply** clauses—Specify the actions to be taken on permitted routes, such as modifying a route attribute.
- **continue** clause—Specifies the next node. A route that matches the current node (permit-mode node) must match the specified next node in the same routing policy. The **continue** clause combines the **if-match** and **apply** clauses of the two nodes to improve flexibility of the routing policy.

Follow these guidelines when you configure **if-match**, **apply**, and **continue** clauses:

- If you only want to filter routes, do not configure **apply** clauses.
- If you do not configure any **if-match** clauses for a permit-mode node, the node will permit all routes.
- Configure a permit-mode node containing no **if-match** or **apply** clauses behind multiple deny-mode nodes to allow unmatched routes to pass.

Configuring filters

Configuration prerequisites

Determine the IP prefix list name, and matching address range.

Configuring an IP prefix list

Configuring an IPv4 prefix list

If all the items are set to **deny** mode, no routes can pass the IPv4 prefix list. To allow unmatched IPv4 routes to pass, you must configure the **permit 0.0.0.0 0 less-equal 32** item following multiple **deny** items.

To configure an IPv4 prefix list:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure an IPv4 prefix list.	ip prefix-list <i>prefix-list-name</i> [index <i>index-number</i>] { deny permit } <i>ip-address mask-length</i> [greater-equal <i>min-mask-length</i>] [less-equal <i>max-mask-length</i>]	By default, no IPv4 prefix list is configured.

Configuring an IPv6 prefix list

If all items are set to **deny** mode, no routes can pass the IPv6 prefix list. To allow unmatched IPv6 routes to pass, you must configure the **permit :: 0 less-equal 128** item following multiple **deny** items.

To configure an IPv6 prefix list:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Configure an IPv6 prefix list.	<ul style="list-style-type: none"> • Method 1: ipv6 prefix-list <i>prefix-list-name</i> [index <i>index-number</i>] { deny permit } <i>ipv6-address prefix-length</i> [greater-equal <i>min-prefix-length</i>] [less-equal <i>max-prefix-length</i>] • Method 2: ipv6 prefix-list <i>prefix-list-name</i> [index 	By default, no IPv6 prefix list is configured. When the inverse keyword is specified, an IPv6 prefix is matched from the least significant

Step	Command	Remarks
	<i>index-number</i>] { deny permit } <i>ipv6-address</i> inverse <i>prefix-length</i>	bit to the most significant bit.

Configuring a routing policy

Configuration prerequisites

Configure filters and routing protocols, and determine the routing policy name, node numbers, match criteria, and the attributes to be modified.

Creating a routing policy

For a routing policy that has more than one node, configure at least one permit-mode node. A route that does not match any node cannot pass the routing policy. If all the nodes are in **deny** mode, no routes can pass the routing policy.

To create a routing policy:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Create a routing policy and a node, and enter routing policy node view.	route-policy <i>route-policy-name</i> { deny permit } node <i>node-number</i>	By default, no routing policy is created.

Configuring if-match clauses

You can either specify no **if-match** clauses or multiple **if-match** clauses for a routing policy node. If no **if-match** clauses are specified for a permit-mode node, all routes can pass the node. If no **if-match** clauses are specified for a deny-mode node, no routes can pass the node.

The **if-match** clauses of a routing policy node have a logical AND relationship. A route must meet all **if-match** clauses before it can be executed by the **apply** clauses of the node. If an **if-match** command exceeds the maximum length, multiple identical **if-match** clauses are generated. These clauses have a logical OR relationship. A route only needs to match one of them.

To configure **if-match** clauses:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter routing policy node view.	route-policy <i>route-policy-name</i> { deny permit } node <i>node-number</i>	N/A

Step	Command	Remarks
3. Match routes whose destination, next hop, or source matches an ACL or prefix list.	<ul style="list-style-type: none"> Match IPv4 routes whose destination, next hop, or source matches an ACL or IPv4 prefix list: if-match ip { address next-hop route-source } { acl <i>acl-number</i> prefix-list <i>prefix-list-name</i> } Match IPv6 routes whose destination, next hop, or source matches an ACL or IPv6 prefix list: if-match ipv6 { address next-hop route-source } { acl <i>acl6-number</i> prefix-list <i>prefix-list-name</i> } 	<p>By default, no ACL or prefix list match criterion is configured.</p> <p>If the ACL used by an if-match clause does not exist, the clause is always matched. If no rules of the specified ACL are matched or the match rules are inactive, the clause is not matched.</p>
4. Match routes having the specified cost.	if-match cost <i>value</i>	By default, no cost match criterion is configured.
5. Match routes having the specified output interface.	if-match interface { <i>interface-type</i> <i>interface-number</i> }&<1-16>	By default, no output interface match criterion is configured.
6. Match IGP routes having the specified tag value.	if-match tag <i>value</i>	By default, no tag match criterion is configured.

Configuring apply clauses

Except for the **apply** commands used for setting the next hop for IPv4 and IPv6 routes, all **apply** commands are the same for IPv4 and IPv6 routing.

To configure **apply** clauses:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter routing policy node view.	route-policy <i>route-policy-name</i> { deny permit } node <i>node-number</i>	N/A
3. Set a cost for routes.	apply cost [+ -] <i>value</i>	By default, no cost is set for routes.
4. Set the next hop for routes.	<ul style="list-style-type: none"> Set the next hop for IPv4 routes: apply ip-address next-hop <i>ip-address</i> [public] Set the next hop for IPv6 routes: apply ipv6 next-hop <i>ipv6-address</i> 	<p>By default, no next hop is set for IPv4/IPv6 routes.</p> <p>The apply ip-address next-hop and apply ipv6 next-hop commands do not apply to redistributed IPv4 and IPv6 routes.</p>
5. Set a preference.	apply preference <i>preference</i>	By default, no preference is set.
6. Set a prefix priority.	apply prefix-priority { critical high medium }	By default, no prefix priority is set, which means the prefix priority is low.
7. Set a tag value for IGP routes.	apply tag <i>value</i>	By default, no tag value is set for IGP routes.

Configuring the continue clause

Follow these guidelines when you configure the **continue** clause to combine multiple nodes:

- If you configure an **apply** clause that sets different attribute values on all the nodes, the **apply** clause on the node configured most recently takes effect.
- If you configure one of the following **apply** clauses on all the nodes, the **apply** clause of each node takes effect:
 - **apply as-path** without the **replace** keyword.
 - **apply cost** with the **+** or **-** keyword.
 - **apply community** with the **additive** keyword.
 - **apply extcommunity** with the **additive** keyword.
- The **apply comm-list delete** clause configured on the current node cannot delete the community attributes set by the **apply community** clauses of the preceding nodes.

To configure the **continue** clause:

Step	Command	Remarks
1. Enter system view.	system-view	N/A
2. Enter routing policy node view.	route-policy <i>route-policy-name</i> { deny permit } node <i>node-number</i>	N/A
3. Specify the next node to be matched.	continue [<i>node-number</i>]	By default, no continue clause is configured. The specified next node must have a larger number than the current node.

Displaying and maintaining the routing policy

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display IPv4 prefix list statistics.	display ip prefix-list [<i>name prefix-list-name</i>]
Display IPv6 prefix list statistics.	display ipv6 prefix-list [<i>name prefix-list-name</i>]
Display routing policy information.	display route-policy [<i>name route-policy-name</i>]
Clear IPv4 prefix list statistics.	reset ip prefix-list [<i>prefix-list-name</i>]
Clear IPv6 prefix list statistics.	reset ipv6 prefix-list [<i>prefix-list-name</i>]

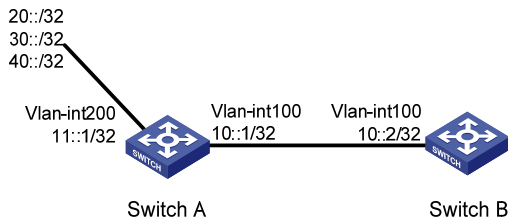
Routing policy configuration example for IPv6 route redistribution

Network requirements

As shown in [Figure 20](#):

- Run RIPng on Switch A and Switch B.
- On Switch A, configure three static routes. Apply a routing policy to static route redistribution to permit routes 20::/32 and 40::/32 and deny route 30::/32.

Figure 20 Network diagram



Configuration procedure

1. Configure Switch A:

Configure IPv6 addresses for VLAN-interface 100 and VLAN-interface 200.

```
<SwitchA> system-view
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ipv6 address 10::1 32
[SwitchA-Vlan-interface100] quit
[SwitchA] interface vlan-interface 200
[SwitchA-Vlan-interface200] ipv6 address 11::1 32
[SwitchA-Vlan-interface200] quit
```

Enable RIPng on VLAN-interface 100.

```
[SwitchA] interface vlan-interface 100
[SwitchA-Vlan-interface100] ripng 1 enable
[SwitchA-Vlan-interface100] quit
```

Configure three static routes with next hop 11::2, and make sure the static routes are active.

```
[SwitchA] ipv6 route-static 20:: 32 11::2
[SwitchA] ipv6 route-static 30:: 32 11::2
[SwitchA] ipv6 route-static 40:: 32 11::2
```

Configure a routing policy.

```
[SwitchA] ipv6 prefix-list a index 10 permit 30:: 32
[SwitchA] route-policy static2ripng deny node 0
[SwitchA-route-policy-static2ripng-0] if-match ipv6 address prefix-list a
[SwitchA-route-policy-static2ripng-0] quit
[SwitchA] route-policy static2ripng permit node 10
[SwitchA-route-policy-static2ripng-10] quit
```

Enable RIPng and apply the routing policy to static route redistribution.

```
[SwitchA] ripng
[SwitchA-ripng-1] import-route static route-policy static2ripng
```

2. Configure Switch B:

Configure the IPv6 address for VLAN-interface 100.

```
<SwitchB> system-view
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ipv6 address 10::2 32
```

Enable RIPng.

```
[SwitchB] ripng
[SwitchB-ripng-1] quit
# Enable RIPng on VLAN-interface 100.
[SwitchB] interface vlan-interface 100
[SwitchB-Vlan-interface100] ripng 1 enable
[SwitchB-Vlan-interface100] quit
```

Verifying the configuration

Display the RIPng routing table on Switch B.

```
[SwitchB] display ripng 1 route
Route Flags: A - Aging, S - Suppressed, G - Garbage-collect
-----

Peer FE80::7D58:0:CA03:1 on Vlan-interface 100
Destination 10::/32,
    via FE80::7D58:0:CA03:1, cost 1, tag 0, A, 18 secs
Destination 20::/32,
    via FE80::7D58:0:CA03:1, cost 1, tag 0, A, 8 secs
Destination 40::/32,
    via FE80::7D58:0:CA03:1, cost 1, tag 0, A, 3 secs
```

Document conventions and icons

Conventions

This section describes the conventions used in the documentation.

Port numbering in examples

The port numbers in this document are for illustration only and might be unavailable on your device.





Command conventions

Convention	Description
Boldface	Bold text represents commands and keywords that you enter literally as shown.
<i>Italic</i>	<i>Italic</i> text represents arguments that you replace with actual values.
[]	Square brackets enclose syntax choices (keywords or arguments) that are optional.
{ x y ... }	Braces enclose a set of required syntax choices separated by vertical bars, from which you select one.
[x y ...]	Square brackets enclose a set of optional syntax choices separated by vertical bars, from which you select one or none.
{ x y ... } *	Asterisk marked braces enclose a set of required syntax choices separated by vertical bars, from which you select at least one.
[x y ...] *	Asterisk marked square brackets enclose optional syntax choices separated by vertical bars, from which you select one choice, multiple choices, or none.
&<1-n>	The argument or keyword and argument combination before the ampersand (&) sign can be entered 1 to n times.
#	A line that starts with a pound (#) sign is comments.













GUI conventions

Convention	Description
Boldface	Window names, button names, field names, and menu items are in Boldface. For example, the New User window appears; click OK .
>	Multi-level menus are separated by angle brackets. For example, File > Create > Folder .

Symbols

Convention	Description
 WARNING!	An alert that calls attention to important information that if not understood or followed can result in personal injury.
 CAUTION:	An alert that calls attention to important information that if not understood or followed can result in data loss, data corruption, or damage to hardware or software.
 IMPORTANT:	An alert that calls attention to essential information.
NOTE:	An alert that contains additional or supplementary information.
 TIP:	An alert that provides helpful information.

Network topology icons

Convention	Description
	Represents a generic network device, such as a router, switch, or firewall.
	Represents a routing-capable device, such as a router or Layer 3 switch.
	Represents a generic switch, such as a Layer 2 or Layer 3 switch, or a router that supports Layer 2 forwarding and other Layer 2 features.
	Represents an access controller, a unified wired-WLAN module, or the access controller engine on a unified wired-WLAN switch.
	Represents an access point.
	Represents a wireless terminator unit.
	Represents a wireless terminator.
	Represents a mesh access point.
	Represents omnidirectional signals.
	Represents directional signals.
	Represents a security product, such as a firewall, UTM, multiservice security gateway, or load balancing device.
	Represents a security card, such as a firewall, load balancing, NetStream, SSL VPN, IPS, or ACG card.

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:
www.hpe.com/assistance
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:
www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates, go to either of the following:
 - Hewlett Packard Enterprise Support Center **Get connected with updates** page:
www.hpe.com/support/e-updates
 - Software Depot website:
www.hpe.com/support/softwaredepot
- To view and update your entitlements, and to link your contracts, Care Packs, and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:
www.hpe.com/support/AccessToSupportMaterials

ⓘ **IMPORTANT:**

Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

Websites

Website	Link
Networking websites	
Hewlett Packard Enterprise Information Library for Networking	www.hpe.com/networking/resourcefinder
Hewlett Packard Enterprise Networking website	www.hpe.com/info/networking
Hewlett Packard Enterprise My Networking website	www.hpe.com/networking/support
Hewlett Packard Enterprise My Networking Portal	www.hpe.com/networking/mynetworking
Hewlett Packard Enterprise Networking Warranty	www.hpe.com/networking/warranty
General websites	
Hewlett Packard Enterprise Information Library	www.hpe.com/info/enterprise/docs
Hewlett Packard Enterprise Support Center	www.hpe.com/support/hpesc
Hewlett Packard Enterprise Support Services Central	ssc.hpe.com/portal/site/ssc/
Contact Hewlett Packard Enterprise Worldwide	www.hpe.com/assistance
Subscription Service/Support Alerts	www.hpe.com/support/e-updates
Software Depot	www.hpe.com/support/softwaredepot
Customer Self Repair (not applicable to all devices)	www.hpe.com/support/selfrepair
Insight Remote Support (not applicable to all devices)	www.hpe.com/info/insightremotesupport/docs

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty, Care Pack Service, or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

For more information and device support details, go to the following website:

www.hpe.com/info/insightremotesupport/docs

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (docsfeedback@hpe.com). When submitting your feedback, include the document title,

part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

Index

A

ACL

- routing policy, [89](#)

action

- IP routing PBR node, [49](#)

advertising

- IP routing RIP default route, [23](#)
- IP routing RIP on interface, [20](#)
- IP routing RIPng default route, [69](#)
- IP routing RIPv2 summary route, [22](#)

applying

- IP routing IPv6 PBR apply clause, [81](#)
- IP routing PBR apply clause, [47](#)
- IP routing RIPng IPsec profile, [72](#)
- routing policy apply clause, [89](#), [92](#)

authenticating

- IP routing RIPng IPsec profile application, [72](#)
- IP routing RIPv2 message authentication configuration, [26](#)

auto

- IP routing RIPv2 automatic route summarization enable, [22](#)

B

backing up

- IP routing route backup, [2](#)

BFD

- IP routing IPv6 static route BFD configuration, [55](#)
- IP routing IPv6 static route BFD control mode (direct next hop), [55](#)
- IP routing IPv6 static route BFD control mode (indirect next hop), [55](#)
- IP routing IPv6 static route BFD echo mode (single hop), [56](#)
- IP routing IPv6 static routing BFD (direct next hop), [59](#)
- IP routing IPv6 static routing BFD (indirect next hop), [61](#)
- IP routing RIP BFD (bidirectional detection/control packet mode), [43](#)
- IP routing RIP BFD (single-hop echo detection), [37](#)
- IP routing RIP BFD (single-hop echo detection/specific destination), [40](#)
- IP routing RIP BFD configuration, [29](#)
- IP routing RIP BFD configuration (bidirectional control detection), [30](#)

- IP routing RIP BFD configuration (single-hop echo detection/neighbor), [29](#)

- IP routing RIP BFD configuration (single-hop echo detection/specific destination), [30](#)

- IP routing static route BFD, [6](#)

- IP routing static route BFD (direct next hop), [10](#)

- IP routing static route BFD (indirect next hop), [12](#)

- IP routing static route BFD bidirectional control mode (direct next hop), [7](#)

- IP routing static route BFD bidirectional control mode (indirect next hop), [7](#)

- IP routing static route BFD single-hop echo mode, [7](#)

bidirectional

- IP routing IPv6 static route BFD control mode (direct next hop), [55](#)

- IP routing IPv6 static route BFD control mode (indirect next hop), [55](#)

- IP routing IPv6 static route BFD echo mode (single hop), [56](#)

- IP routing RIP BFD (bidirectional detection/control packet mode), [43](#)

- IP routing RIP BFD configuration (bidirectional control detection), [30](#)

- IP routing static route BFD bidirectional control mode (direct next hop), [7](#)

- IP routing static route BFD bidirectional control mode (indirect next hop), [7](#)

C

configuring

- IP routing, [1](#)

- IP routing FIB route max lifetime, [3](#)

- IP routing IPv6 default route, [65](#)

- IP routing IPv6 PBR, [81](#), [82](#), [83](#), [84](#)

- IP routing IPv6 PBR interface, [84](#)

- IP routing IPv6 PBR interface (packet type-based), [86](#)

- IP routing IPv6 PBR local, [83](#)

- IP routing IPv6 PBR local (packet type-based), [84](#)

- IP routing IPv6 PBR node action, [83](#)

- IP routing IPv6 PBR node match criteria, [82](#)

- IP routing IPv6 PBR policy, [82](#)

- IP routing IPv6 static route, [55](#)

- IP routing IPv6 static route BFD, [55](#)

- IP routing IPv6 static route BFD control mode (direct next hop), [55](#)

- IP routing IPv6 static route BFD control mode (indirect next hop), [55](#)

- IP routing IPv6 static route BFD echo mode (single hop), 56
- IP routing IPv6 static routing, 55, 57
- IP routing IPv6 static routing basics, 57
- IP routing IPv6 static routing BFD (direct next hop), 59
- IP routing IPv6 static routing BFD (indirect next hop), 61
- IP routing PBR, 47, 48, 49, 50
- IP routing PBR (interface), 50
- IP routing PBR (interface/packet type-based), 52
- IP routing PBR (local), 49
- IP routing PBR (local/packet type-based), 50
- IP routing PBR node action, 49
- IP routing PBR node match criteria, 48
- IP routing PBR policy, 48
- IP routing RIB label max lifetime, 3
- IP routing RIB route max lifetime, 3
- IP routing RIP, 17, 18, 31
- IP routing RIP additional routing metric, 21
- IP routing RIP basics, 19, 31
- IP routing RIP BFD, 29
- IP routing RIP BFD (bidirectional control detection), 30
- IP routing RIP BFD (bidirectional detection/control packet mode), 43
- IP routing RIP BFD (single-hop echo detection), 37
- IP routing RIP BFD (single-hop echo detection/neighbor), 29
- IP routing RIP BFD (single-hop echo detection/specific destination), 30, 40
- IP routing RIP GR, 28
- IP routing RIP interface additional metric, 36
- IP routing RIP network management, 27
- IP routing RIP packet send rate, 28
- IP routing RIP poison reverse, 25
- IP routing RIP preference, 24
- IP routing RIP received/redistributed route filtering, 23
- IP routing RIP route control, 21
- IP routing RIP route redistribution, 24, 34
- IP routing RIP split horizon, 25
- IP routing RIP timers, 24
- IP routing RIP version, 20
- IP routing RIPng, 66, 67, 73
- IP routing RIPng basics, 67, 73
- IP routing RIPng GR, 71
- IP routing RIPng IPsec profile configuration, 78
- IP routing RIPng packet zero field check, 71
- IP routing RIPng poison reverse, 70
- IP routing RIPng preference, 69
- IP routing RIPng received/redistributed route filtering, 69
- IP routing RIPng route control, 68
- IP routing RIPng route redistribution, 70, 75
- IP routing RIPng route summarization, 68
- IP routing RIPng routing metric, 68
- IP routing RIPng split horizon, 70
- IP routing RIPng timer, 70
- IP routing RIPv2 message authentication, 26
- IP routing RIPv2 route summarization, 21
- IP routing static configuration, 6
- IP routing static default route, 16
- IP routing static route, 6, 8
- IP routing static route basics, 8
- IP routing static route BFD, 6
- IP routing static route BFD (direct next hop), 10
- IP routing static route BFD (indirect next hop), 12
- IP routing static route BFD bidirectional control mode (direct next hop), 7
- IP routing static route BFD bidirectional control mode (indirect next hop), 7
- IP routing static route BFD single-hop echo mode, 7
- routing policy, 89, 91
- routing policy (IPv6 route redistribution), 93
- routing policy apply clause, 92
- routing policy continue clause, 93
- routing policy filter, 90
- routing policy if-match clause, 91
- routing policy IPv4 prefix list, 90
- routing policy IPv6 prefix list, 90
- continue clause (routing policy), 89, 93
- controlling
 - IP routing IPv6 static route BFD control mode (direct next hop), 55
 - IP routing IPv6 static route BFD control mode (indirect next hop), 55
 - IP routing RIP additional routing metric configuration, 21
 - IP routing RIP BFD configuration (bidirectional detection/control packet mode), 43
 - IP routing RIP interface advertisement, 20
 - IP routing RIP interface reception, 20, 20
 - IP routing RIP route control configuration, 21
 - IP routing RIPng route control, 68
- creating
 - IP routing IPv6 PBR node, 82
 - IP routing PBR node, 48
 - routing policy, 91

D

default

- IP routing IPv6 default route configuration, 65
- IP routing RIP default route advertisement, 23
- IP routing RIPng default route advertisement, 69
- IP routing static route configuration. See under static routing

detecting

- IP routing RIP BFD (bidirectional detection/control packet mode), 43
- IP routing RIP BFD (single-hop echo detection), 37
- IP routing RIP BFD (single-hop echo detection/specific destination), 40
- IP routing RIP BFD configuration (bidirectional control detection), 30
- IP routing RIP BFD configuration (single-hop echo detection/neighbor), 29
- IP routing RIP BFD configuration (single-hop echo detection/specific destination), 30
- IP routing RIP BFD single-hop echo detection, 29

device

- routing policy configuration (IPv6 route redistribution), 93

disabling

- IP routing RIP host route reception, 22

displaying

- IP routing IPv6 PBR, 84
- IP routing IPv6 static routing, 57
- IP routing PBR, 50
- IP routing RIP, 30
- IP routing RIPng, 72
- IP routing static routes, 8
- IP routing table, 4
- routing policy, 93

distributing

- IP routing RIPng received/redistributed route filtering, 69
- IP routing RIPng route redistribution, 70, 75
- IP routing route redistribution, 3

dynamic

- IP routing dynamic routing protocols, 2

E

echo

- IP routing IPv6 static route BFD echo mode (single hop), 56
- IP routing RIP BFD (bidirectional detection/control packet mode), 43

IP routing RIP BFD (single-hop echo detection), 37

IP routing RIP BFD (single-hop echo detection/specific destination), 40

IP routing RIP BFD single-hop echo detection, 29

IP routing static route BFD single-hop echo mode, 7

enabling

- IP routing RIP, 19
- IP routing RIP (interface), 19
- IP routing RIP (network), 19
- IP routing RIP poison reverse, 25
- IP routing RIP split horizon, 25
- IP routing RIP update source IP address check, 26
- IP routing RIPv1 incoming message zero field check, 26
- IP routing RIPv2 automatic route summarization, 22

F

FIB

- IP routing table, 1

filtering

- IP routing RIP received/redistributed route filtering, 23
- IP routing RIPng received/redistributed route filtering, 69
- routing policy ACLs, 89
- routing policy apply clause, 92
- routing policy configuration, 89, 91
- routing policy configuration (IPv6 route redistribution), 93
- routing policy continue clause, 93
- routing policy creation, 91
- routing policy filter configuration, 90
- routing policy filters, 89
- routing policy if-match clause, 91
- routing policy IP prefix list, 90
- routing policy prefix list, 89

forwarding

- IP routing IPv6 PBR configuration, 81, 82, 83, 84
- IP routing IPv6 PBR interface configuration, 84
- IP routing IPv6 PBR interface configuration (packet type-based), 86
- IP routing IPv6 PBR local configuration, 83
- IP routing IPv6 PBR local configuration (packet type-based), 84
- IP routing IPv6 PBR policy configuration, 82
- IP routing PBR configuration, 47, 48, 49, 50
- IP routing PBR configuration (interface), 50

- IP routing PBR configuration (interface/packet type-based), [52](#)
- IP routing PBR configuration (local), [49](#)
- IP routing PBR configuration (local/packet type-based), [50](#)
- IP routing PBR policy configuration, [48](#)

G

- garbage-collect timer (RIP), [24](#)
- GR helper
 - IP routing RIP GR helper configuration, [28](#)
 - IP routing RIPng, [71](#)
- GR restarter
 - IP routing RIP GR restarter configuration, [28](#)
 - IP routing RIPng, [71](#)
- Graceful Restart (GR)
 - IP routing RIP configuration, [28](#)
 - IP routing RIPng configuration, [71](#)

H

- hop
 - IP routing RIP BFD configuration (bidirectional control detection), [30](#)
 - IP routing RIP BFD configuration (single-hop echo detection/neighbor), [29](#)
 - IP routing RIP BFD configuration (single-hop echo detection/specific destination), [30](#)
- host route reception, [22](#)

I

- IGP
 - IP routing RIP BFD configuration (bidirectional control detection), [30](#)
 - IP routing RIP BFD configuration (single-hop echo detection/neighbor), [29](#)
 - IP routing RIP BFD configuration (single-hop echo detection/specific destination), [30](#)
 - IP routing RIP configuration, [17](#), [18](#), [31](#)
 - IP routing RIP neighbor specification, [27](#)
- interface
 - IP routing IPv6 PBR interface configuration, [84](#)
 - IP routing IPv6 PBR interface configuration (packet type-based), [86](#)
 - IP routing PBR configuration (interface), [50](#)
 - IP routing PBR configuration (interface/packet type-based), [52](#)
- IP addressing
 - IP routing RIP configuration, [17](#), [31](#)
 - IP routing RIP update source IP address check, [26](#)
 - RIP configuration, [18](#)
- IP routing

- configuration, [1](#)
- displaying IPv6 static routing, [57](#)
- displaying PBR, [50](#)
- displaying RIPng, [72](#)
- displaying routing table, [4](#)
- displaying static routes, [8](#)
- dynamic routing protocols, [2](#)
- FIB route max lifetime, [3](#)
- IP routing static route BFD bidirectional control mode (indirect next hop), [7](#)
- IPv6 default route. See under IPv6 static routing
- IPv6 policy-based routing. See [IPv6 PBR](#)
- IPv6 static routing. See [IPv6 static routing](#)
- maintaining PBR, [50](#)
- maintaining RIPng, [72](#)
- maintaining routing table, [4](#)
- PBR configuration, [48](#), [49](#), [50](#)
- PBR configuration (interface), [50](#)
- PBR configuration (local), [49](#)
- PBR node action, [49](#)
- PBR node creation, [48](#)
- PBR node match criteria, [48](#)
- PBR policy, [47](#)
- PBR policy configuration, [48](#)
- PBR-Track collaboration, [48](#)
- policy apply clause, [92](#)
- policy configuration, [89](#), [91](#)
- policy configuration (IPv6 route redistribution), [93](#)
- policy continue clause, [93](#)
- policy creation, [91](#)
- policy display, [93](#)
- policy filter configuration, [90](#)
- policy filtering, [89](#)
- policy filters, [89](#)
- policy if-match clause, [91](#)
- policy IP prefix list, [90](#)
- policy maintain, [93](#)
- policy-based routing. Use [PBR](#)
- RIB label max lifetime, [3](#)
- RIB route max lifetime, [3](#)
- RIP additional routing metric configuration, [21](#)
- RIP basic configuration, [19](#)
- RIP basics, [31](#)
- RIP BFD (bidirectional detection/control packet mode), [43](#)
- RIP BFD (single-hop echo detection), [37](#)
- RIP BFD (single-hop echo detection/specific destination), [40](#)
- RIP BFD configuration, [29](#)
- RIP BFD configuration (bidirectional control detection), [30](#)

- RIP BFD configuration (single-hop echo detection/neighbor), 29
- RIP BFD configuration (single-hop echo detection/specific destination), 30
- RIP configuration, 17, 18, 31
- RIP default route advertisement, 23
- RIP GR configuration, 28
- RIP host route reception disable, 22
- RIP interface additional metric, 36
- RIP interface advertisement control, 20
- RIP interface reception control, 20
- RIP neighbor specification, 27
- RIP network management configuration, 27
- RIP network optimization, 24
- RIP network tuning, 24
- RIP operation, 17
- RIP packet max length, 28
- RIP packet send rate configuration, 28
- RIP poison reverse configuration, 25
- RIP preference configuration, 24
- RIP received/redistributed route filtering, 23
- RIP route control configuration, 21
- RIP route entries, 17
- RIP route redistribution, 34
- RIP route redistribution configuration, 24
- RIP routing loop prevention, 17
- RIP split horizon configuration, 25
- RIP timer configuration, 24
- RIP update source IP address check, 26
- RIP version configuration, 20
- RIP versions, 18
- RIPng basic configuration, 67, 73
- RIPng configuration, 66, 67, 73
- RIPng default route advertisement, 69
- RIPng GR configuration, 71
- RIPng IPsec profile application, 72
- RIPng IPsec profile configuration, 78
- RIPng network optimization, 70
- RIPng network tuning, 70
- RIPng packet, 66
- RIPng packet zero field check configuration, 71
- RIPng poison reverse configuration, 70
- RIPng preference, 69
- RIPng protocols and standards, 67
- RIPng received/redistributed route filtering, 69
- RIPng route control, 68
- RIPng route entry, 66
- RIPng route redistribution, 70, 75
- RIPng route summarization, 68
- RIPng routing metric configuration, 68
- RIPng split horizon configuration, 70
- RIPng timer configuration, 70
- RIPv1 message zero field check, 26
- RIPv2 message authentication configuration, 26
- RIPv2 route summarization configuration, 21
- route backup, 2
- route preference, 2
- route recursion, 2
- route redistribution, 3
- routing table, 1
- static default route, 16
- static route, 6, 8
- static route basics, 8
- static route BFD, 6
- static route BFD (direct next hop), 10
- static route BFD (indirect next hop), 12
- static route BFD bidirectional control mode (direct next hop), 7
- static route BFD single-hop echo mode, 7
- static routing configuration, 6
- IPsec
 - IP routing RIPng IPsec profile application, 72
 - IP routing RIPng IPsec profile configuration, 78
- IPv4
 - IP routing FIB route max lifetime, 3
 - IP routing RIB label max lifetime, 3
 - IP routing RIB route max lifetime, 3
 - routing policy ACLs, 89
 - routing policy configuration, 89
 - routing policy IP prefix list, 90
 - routing policy prefix list, 89
- IPv6
 - IP routing FIB route max lifetime, 3
 - IP routing RIB label max lifetime, 3
 - IP routing RIB route max lifetime, 3
 - policy-based routing. See [IPv6 PBR](#)
 - RIP, 66, See also [RIPng](#)
 - routing policy ACLs, 89
 - routing policy configuration, 89
 - routing policy configuration (IPv6 route redistribution), 93
 - routing policy IP prefix list, 90
 - routing policy prefix list, 89
- IPv6 PBR
 - apply clause, 81
 - configuration, 81, 82, 83, 84
 - displaying, 84
 - if-match clause, 81
 - interface configuration, 84
 - interface configuration (packet type-based), 86
 - interface PBR, 81

- local configuration, [83](#)
- local configuration (packet type-based), [84](#)
- local PBR, [81](#)
- maintaining, [84](#)
- match mode/node clause relationship, [81](#)
- node action, [83](#)
- node creation, [82](#)
- node match criteria, [82](#)
- policy, [81](#)
- policy configuration, [82](#)
- Track collaboration, [82](#)

- IPv6 static routing
 - basic configuration, [57](#)
 - BFD configuration, [55](#)
 - BFD configuration (direct next hop), [59](#)
 - BFD configuration (indirect next hop), [61](#)
 - BFD control mode (direct next hop), [55](#)
 - BFD control mode (indirect next hop), [55](#)
 - BFD echo mode (single hop), [56](#)
 - configuration, [55](#), [57](#)
 - default route configuration, [65](#)
 - displaying, [57](#)
 - route configuration, [55](#)

L

- label
 - IP routing RIB label max lifetime, [3](#)
- list
 - routing policy IP prefix list, [90](#)
 - routing policy prefix list, [89](#)

- local
 - IP routing IPv6 PBR local configuration, [83](#)
 - IP routing IPv6 PBR local configuration (packet type-based), [84](#)
 - IP routing PBR configuration (local), [49](#)
 - IP routing PBR configuration (local/packet type-based), [50](#)

- loop
 - IP routing RIP routing loop prevention, [17](#)

M

- maintaining
 - IP routing IPv6 PBR, [84](#)
 - IP routing PBR, [50](#)
 - IP routing RIP, [30](#)
 - IP routing RIPng, [72](#)
 - IP routing table, [4](#)
 - routing policy, [93](#)

- matching
 - IP routing IPv6 PBR if-match clause, [81](#)
 - IP routing IPv6 PBR node action, [83](#)

- IP routing IPv6 PBR node match criteria, [82](#)
- IP routing PBR deny match mode, [47](#)
- IP routing PBR if-match clause, [47](#)
- IP routing PBR node match criteria, [48](#)
- IP routing PBR permit match mode, [47](#)
- routing policy if-match clause, [89](#), [91](#)

- message

- IP routing RIPv1 message zero field check enable, [26](#)
- IP routing RIPv2 message authentication configuration, [26](#)

- metric

- IP routing RIP additional routing metric configuration, [21](#)
- IP routing RIP interface additional metric, [36](#)
- IP routing RIPng routing metric configuration, [68](#)

- mode

- IP routing IPv6 static route BFD control mode (direct next hop), [55](#)
- IP routing IPv6 static route BFD control mode (indirect next hop), [55](#)
- IP routing IPv6 static route BFD echo mode (single hop), [56](#)
- IP routing PBR deny match mode, [47](#)
- IP routing PBR permit match mode, [47](#)
- IP routing static route BFD bidirectional control mode (direct next hop), [7](#)
- IP routing static route BFD bidirectional control mode (indirect next hop), [7](#)
- IP routing static route BFD single-hop echo mode, [7](#)

- multicast

- IP routing RIPng basic configuration, [67](#)
- IP routing RIPng configuration, [66](#), [67](#)

N

- neighbor

- IP routing RIP neighbor specification, [27](#)

- network

- IP routing dynamic routing protocols, [2](#)
- IP routing FIB route max lifetime, [3](#)
- IP routing IPv6 PBR configuration, [83](#)
- IP routing IPv6 PBR interface configuration, [84](#)
- IP routing IPv6 PBR local configuration, [83](#)
- IP routing IPv6 PBR node action, [83](#)
- IP routing IPv6 PBR node creation, [82](#)
- IP routing IPv6 PBR node match criteria, [82](#)
- IP routing IPv6 PBR policy configuration, [82](#)
- IP routing IPv6 PBR-Track collaboration, [82](#)
- IP routing IPv6 static route BFD configuration, [55](#)
- IP routing IPv6 static route BFD control mode (direct next hop), [55](#)

- IP routing IPv6 static route BFD control mode (indirect next hop), [55](#)
- IP routing IPv6 static route BFD echo mode (single hop), [56](#)
- IP routing IPv6 static route configuration, [55](#)
- IP routing PBR node action, [49](#)
- IP routing PBR node creation, [48](#)
- IP routing PBR node match criteria, [48](#)
- IP routing PBR policy, [47](#)
- IP routing PBR policy configuration, [48](#)
- IP routing PBR-Track collaboration, [48](#)
- IP routing RIB label max lifetime, [3](#)
- IP routing RIB route max lifetime, [3](#)
- IP routing RIP additional routing metric configuration, [21](#)
- IP routing RIP basic configuration, [19](#)
- IP routing RIP basics, [31](#)
- IP routing RIP BFD (bidirectional detection/control packet mode), [43](#)
- IP routing RIP BFD (single-hop echo detection), [37](#)
- IP routing RIP BFD (single-hop echo detection/specific destination), [40](#)
- IP routing RIP BFD configuration, [29](#)
- IP routing RIP default route advertisement, [23](#)
- IP routing RIP GR configuration, [28](#)
- IP routing RIP host route reception disable, [22](#)
- IP routing RIP interface additional metric, [36](#)
- IP routing RIP interface advertisement control, [20](#)
- IP routing RIP interface reception control, [20](#)
- IP routing RIP network management configuration, [27](#)
- IP routing RIP network optimization, [24](#)
- IP routing RIP network tuning, [24](#)
- IP routing RIP operation, [17](#)
- IP routing RIP packet max length, [28](#)
- IP routing RIP packet send rate configuration, [28](#)
- IP routing RIP poison reverse configuration, [25](#)
- IP routing RIP preference configuration, [24](#)
- IP routing RIP received/redistributed route filtering, [23](#)
- IP routing RIP route control configuration, [21](#)
- IP routing RIP route entries, [17](#)
- IP routing RIP route redistribution, [34](#)
- IP routing RIP route redistribution configuration, [24](#)
- IP routing RIP routing loop prevention, [17](#)
- IP routing RIP split horizon configuration, [25](#)
- IP routing RIP timer configuration, [24](#)

- IP routing RIP update source IP address check, [26](#)
- IP routing RIP version configuration, [20](#)
- IP routing RIP versions, [18](#)
- IP routing RIPng basic configuration, [67](#)
- IP routing RIPng default route advertisement, [69](#)
- IP routing RIPng GR configuration, [71](#)
- IP routing RIPng IPsec profile application, [72](#)
- IP routing RIPng network optimization, [70](#)
- IP routing RIPng network tuning, [70](#)
- IP routing RIPng packet, [66](#)
- IP routing RIPng packet zero field check, [71](#)
- IP routing RIPng poison reverse, [70](#)
- IP routing RIPng preference, [69](#)
- IP routing RIPng received/redistributed route filtering, [69](#)
- IP routing RIPng route control, [68](#)
- IP routing RIPng route entry, [66](#)
- IP routing RIPng route redistribution, [70](#)
- IP routing RIPng route summarization, [68](#)
- IP routing RIPng routing metric configuration, [68](#)
- IP routing RIPng split horizon, [70](#)
- IP routing RIPng timer configuration, [70](#)
- IP routing RIPv1 message zero field check, [26](#)
- IP routing RIPv2 message authentication configuration, [26](#)
- IP routing RIPv2 route summarization configuration, [21](#)
- IP routing route backup, [2](#)
- IP routing route preference, [2](#)
- IP routing route recursion, [2](#)
- IP routing route redistribution, [3](#)
- IP routing static route, [6](#)
- IP routing static route BFD, [6](#)
- IP routing static route BFD bidirectional control mode (direct next hop), [7](#)
- IP routing static route BFD bidirectional control mode (indirect next hop), [7](#)
- IP routing static route BFD single-hop echo mode, [7](#)
- routing policy apply clause, [92](#)
- routing policy configuration, [91](#)
- routing policy configuration (IPv6 route redistribution), [93](#)
- routing policy continue clause, [93](#)
- routing policy creation, [91](#)
- routing policy filter configuration, [90](#)
- routing policy if-match clause, [91](#)
- routing policy IP prefix list, [90](#)
- network management
 - IP routing configuration, [1](#)

- IP routing IPv6 default route configuration, 65
- IP routing IPv6 PBR configuration, 81, 82, 84
- IP routing IPv6 PBR interface configuration (packet type-based), 86
- IP routing IPv6 PBR local configuration (packet type-based), 84
- IP routing IPv6 static routing basic configuration, 57
- IP routing IPv6 static routing BFD (direct next hop), 59
- IP routing IPv6 static routing BFD (indirect next hop), 61
- IP routing IPv6 static routing configuration, 55, 57
- IP routing PBR configuration, 47, 48, 49, 50
- IP routing PBR configuration (interface/packet type-based), 52
- IP routing PBR configuration (local/packet type-based), 50
- IP routing RIP configuration, 17, 18, 31
- IP routing RIPng basic configuration, 73
- IP routing RIPng configuration, 66, 67, 73
- IP routing RIPng IPsec profile configuration, 78
- IP routing RIPng route redistribution, 75
- IP routing static configuration, 6
- IP routing static default route, 16
- IP routing static route, 8
- IP routing static route basics, 8
- IP routing static route BFD (direct next hop), 10
- IP routing static route BFD (indirect next hop), 12
- routing policy configuration, 89

node

- IP routing IPv6 PBR node action, 83
- IP routing IPv6 PBR node creation, 82
- IP routing IPv6 PBR node match criteria configuration, 82
- IP routing IPv6 PBR policy, 81
- IP routing IPv6 PBR-Track collaboration, 82
- IP routing PBR apply clause, 47
- IP routing PBR creation, 48
- IP routing PBR if-match clause, 47
- IP routing PBR match criteria, 48
- IP routing PBR node action, 49
- IP routing PBR policy, 47
- IP routing PBR-Track collaboration, 48
- routing policy apply clause, 89, 92
- routing policy continue clause, 89, 93
- routing policy deny match, 89
- routing policy if-match clause, 89, 91

routing policy permit match, 89

O

optimal

- IP routing FIB table optimal routes, 1

optimizing

- IP routing RIP networks, 24
- IP routing RIPng network, 70

P

packet

- IP routing configuration, 1
- IP routing dynamic routing protocols, 2
- IP routing IPv6 PBR configuration, 81, 82, 83, 84
- IP routing IPv6 PBR interface configuration, 84
- IP routing IPv6 PBR interface configuration (packet type-based), 86
- IP routing IPv6 PBR local configuration, 83
- IP routing IPv6 PBR local configuration (packet type-based), 84
- IP routing IPv6 PBR policy, 81
- IP routing IPv6 PBR policy configuration, 82
- IP routing PBR configuration, 47, 48, 49, 50
- IP routing PBR configuration (interface), 50
- IP routing PBR configuration (interface/packet type-based), 52
- IP routing PBR configuration (local), 49
- IP routing PBR configuration (local/packet type-based), 50
- IP routing PBR policy configuration, 48
- IP routing RIP BFD configuration (bidirectional control detection), 30
- IP routing RIP BFD configuration (single-hop echo detection/neighbor), 29
- IP routing RIP BFD configuration (single-hop echo detection/specific destination), 30
- IP routing RIP network management configuration, 27
- IP routing RIP packet max length, 28
- IP routing RIP packet send rate configuration, 28
- IP routing RIPng, 66
- IP routing RIPng packet zero field check, 71
- IP routing route backup, 2
- IP routing route preference, 2
- IP routing route recursion, 2
- IP routing route redistribution, 3

PBR

- configuration, 47, 48, 49, 50
- configuration (interface), 50
- configuration (interface/packet type-based), 52
- configuration (local), 49
- configuration (local/packet type-based), 50

- displaying, [50](#)
- interface PBR, [47](#)
- local PBR, [47](#)
- maintaining, [50](#)
- node action configuration, [49](#)
- node creation, [48](#)
- node match criteria, [48](#)
- policy, [47](#)
- policy configuration, [48](#)
- relationship between match mode/clauses, [47](#)
- Track collaboration, [48](#)

poison reverse

- configuration, [25](#)
- enable, [25](#)
- IP routing RIP routing loop prevention, [17](#)
- IP routing RIPng configuration, [70](#)

policy

- IP routing IPv6 PBR, [81](#)
- IP routing IPv6 PBR apply clause, [81](#)
- IP routing IPv6 PBR configuration, [81](#), [82](#), [83](#), [84](#)
- IP routing IPv6 PBR if-match clause, [81](#)
- IP routing IPv6 PBR interface configuration, [84](#)
- IP routing IPv6 PBR interface configuration (packet type-based), [86](#)
- IP routing IPv6 PBR local configuration, [83](#)
- IP routing IPv6 PBR local configuration (packet type-based), [84](#)
- IP routing IPv6 PBR match mode/node clause relationship, [81](#)
- IP routing IPv6 PBR policy configuration, [82](#)
- IP routing PBR, [47](#)
- IP routing PBR configuration, [47](#), [48](#), [48](#), [49](#), [50](#)
- IP routing PBR configuration (interface), [50](#)
- IP routing PBR configuration (interface/packet type-based), [52](#)
- IP routing PBR configuration (local), [49](#)
- IP routing PBR configuration (local/packet type-based), [50](#)
- IP routing PBR node action, [49](#)
- IP routing PBR node creation, [48](#)
- IP routing PBR node match criteria, [48](#)
- routing policy apply clause, [92](#)
- routing policy configuration, [89](#), [91](#)
- routing policy configuration (IPv6 route redistribution), [93](#)
- routing policy continue clause, [93](#)
- routing policy creation, [91](#)
- routing policy filter configuration, [90](#)
- routing policy filtering, [89](#)
- routing policy if-match clause, [91](#)
- routing policy IP prefix list, [90](#)

policy-based routing. Use [PBR](#)

preference

- IP routing RIP configuration, [24](#)
- IP routing RIPng preference, [69](#)
- IP routing route preference, [2](#)

prefix

- routing policy IP prefix list, [90](#)
- routing policy prefix list, [89](#)

procedure

- advertising IP routing RIP default route, [23](#)
- advertising IP routing RIPng default route, [69](#)
- advertising IP routing RIPv2 summary route, [22](#)
- applying IP routing RIPng IPsec profile, [72](#)
- configuring IP routing FIB route max lifetime, [3](#)
- configuring IP routing IPv6 PBR, [82](#), [83](#), [84](#)
- configuring IP routing IPv6 PBR interface, [84](#)
- configuring IP routing IPv6 PBR interface (packet type-based), [86](#)
- configuring IP routing IPv6 PBR local, [83](#)
- configuring IP routing IPv6 PBR local (packet type-based), [84](#)
- configuring IP routing IPv6 PBR node action, [83](#)
- configuring IP routing IPv6 PBR node match criteria, [82](#)
- configuring IP routing IPv6 PBR policy, [82](#)
- configuring IP routing IPv6 static route, [55](#)
- configuring IP routing IPv6 static route BFD, [55](#)
- configuring IP routing IPv6 static route BFD control mode (direct next hop), [55](#)
- configuring IP routing IPv6 static route BFD control mode (indirect next hop), [55](#)
- configuring IP routing IPv6 static route BFD echo mode (single hop), [56](#)
- configuring IP routing IPv6 static routing, [57](#)
- configuring IP routing IPv6 static routing basics, [57](#)
- configuring IP routing IPv6 static routing BFD (direct next hop), [59](#)
- configuring IP routing IPv6 static routing BFD (indirect next hop), [61](#)
- configuring IP routing PBR, [48](#), [49](#), [50](#)
- configuring IP routing PBR (interface), [50](#)
- configuring IP routing PBR (interface/packet type-based), [52](#)
- configuring IP routing PBR (local), [49](#)
- configuring IP routing PBR (local/packet type-based), [50](#)
- configuring IP routing PBR node action, [49](#)
- configuring IP routing PBR node match criteria, [48](#)

- configuring IP routing PBR policy, [48](#)
- configuring IP routing RIB label max lifetime, [3](#)
- configuring IP routing RIB route max lifetime, [3](#)
- configuring IP routing RIP, [18](#), [31](#)
- configuring IP routing RIP additional routing metric, [21](#)
- configuring IP routing RIP basics, [19](#), [31](#)
- configuring IP routing RIP BFD, [29](#)
- configuring IP routing RIP BFD (bidirectional control detection), [30](#)
- configuring IP routing RIP BFD (bidirectional detection/control packet mode), [43](#)
- configuring IP routing RIP BFD (single-hop echo detection), [37](#)
- configuring IP routing RIP BFD (single-hop echo detection/neighbor), [29](#)
- configuring IP routing RIP BFD (single-hop echo detection/specific destination), [30](#), [40](#)
- configuring IP routing RIP GR, [28](#)
- configuring IP routing RIP interface additional metric, [36](#)
- configuring IP routing RIP network management, [27](#)
- configuring IP routing RIP packet send rate, [28](#)
- configuring IP routing RIP poison reverse, [25](#)
- configuring IP routing RIP preference, [24](#)
- configuring IP routing RIP received/redistributed route filtering, [23](#)
- configuring IP routing RIP route control, [21](#)
- configuring IP routing RIP route redistribution, [24](#), [34](#)
- configuring IP routing RIP split horizon, [25](#)
- configuring IP routing RIP timers, [24](#)
- configuring IP routing RIP version, [20](#)
- configuring IP routing RIPng, [67](#), [73](#)
- configuring IP routing RIPng basics, [67](#), [73](#)
- configuring IP routing RIPng GR, [71](#)
- configuring IP routing RIPng IPsec profile configuration, [78](#)
- configuring IP routing RIPng packet zero field check, [71](#)
- configuring IP routing RIPng poison reverse, [70](#)
- configuring IP routing RIPng preference, [69](#)
- configuring IP routing RIPng received/redistributed route filtering, [69](#)
- configuring IP routing RIPng route control, [68](#)
- configuring IP routing RIPng route redistribution, [70](#), [75](#)
- configuring IP routing RIPng route summarization, [68](#)
- configuring IP routing RIPng split horizon, [70](#)
- configuring IP routing RIPng timer, [70](#)
- configuring IP routing RIPv2 message authentication, [26](#)
- configuring IP routing RIPv2 route summarization, [21](#)
- configuring IP routing static route, [6](#), [8](#)
- configuring IP routing static route basics, [8](#)
- configuring IP routing static route BFD, [6](#)
- configuring IP routing static route BFD (direct next hop), [10](#)
- configuring IP routing static route BFD (indirect next hop), [12](#)
- configuring IP routing static route BFD bidirectional control mode (direct next hop), [7](#)
- configuring IP routing static route BFD bidirectional control mode (indirect next hop), [7](#)
- configuring IP routing static route BFD single-hop echo mode, [7](#)
- configuring IP routing static route default route, [16](#)
- configuring RIPng routing metric, [68](#)
- configuring routing policy, [91](#)
- configuring routing policy (IPv6 route redistribution), [93](#)
- configuring routing policy apply clause, [92](#)
- configuring routing policy continue clause, [93](#)
- configuring routing policy filter, [90](#)
- configuring routing policy if-match clause, [91](#)
- configuring routing policy IPv4 prefix list, [90](#)
- configuring routing policy IPv6 prefix list, [90](#)
- controlling IP routing RIP interface advertisement, [20](#)
- controlling IP routing RIP interface reception, [20](#)
- creating IP routing IPv6 PBR node, [82](#)
- creating IP routing PBR node, [48](#)
- creating routing policy, [91](#)
- disabling IP routing RIP host route reception, [22](#)
- displaying IP routing IPv6 PBR, [84](#)
- displaying IP routing IPv6 static routing, [57](#)
- displaying IP routing PBR, [50](#)
- displaying IP routing RIP, [30](#)
- displaying IP routing RIPng, [72](#)
- displaying IP routing static routes, [8](#)
- displaying IP routing table, [4](#)
- displaying routing policy, [93](#)
- enabling IP routing RIP, [19](#)
- enabling IP routing RIP (interface), [19](#)
- enabling IP routing RIP (network), [19](#)
- enabling IP routing RIP poison reverse, [25](#)
- enabling IP routing RIP split horizon, [25](#)
- enabling IP routing RIP update source IP address check, [26](#)

- enabling IP routing RIPv1 message zero field check, 26
- enabling IP routing RIPv2 automatic route summarization, 22
- maintaining IP routing IPv6 PBR, 84
- maintaining IP routing PBR, 50
- maintaining IP routing RIP, 30
- maintaining IP routing RIPng, 72
- maintaining IP routing table, 4
- maintaining routing policy, 93
- optimizing IP routing RIP networks, 24
- optimizing IP routing RIPng network, 70
- setting IP routing RIP packet max length, 28
- specifying IP routing RIP neighbor, 27
- tuning IP routing RIP networks, 24
- tuning IP routing RIPng network, 70
- protocols and standards
 - IP routing dynamic routing protocols, 2
 - IP routing RIP, 18
 - IP routing RIPng, 67
- R**
- recursion
 - IP routing route recursion, 2
- redistributing
 - IP routing RIP received/redistributed route filtering, 23
 - IP routing RIP route redistribution, 34
 - IP routing RIP routes, 24
 - IP routing RIPng received/redistributed route filtering, 69
 - IP routing RIPng route redistribution, 70, 75
 - IP routing route redistribution, 3
- RIB
 - IP routing FIB route max lifetime, 3
 - IP routing RIB label max lifetime, 3
 - IP routing RIB route max lifetime, 3
- RIP, 66, *See also* RIPng
 - additional routing metric configuration, 21
 - basic configuration, 19, 31
 - BFD configuration, 29
 - BFD configuration (bidirectional control detection), 30
 - BFD configuration (bidirectional detection/control packet mode), 43
 - BFD configuration (single-hop echo detection), 37
 - BFD configuration (single-hop echo detection/neighbor), 29
 - BFD configuration (single-hop echo detection/specific destination), 30, 40
 - configuration, 17, 18, 31
 - default route advertisement, 23
 - displaying, 30
 - enabling, 19
 - GR configuration, 28
 - GR helper configuration, 28
 - GR restarter configuration, 28
 - host route reception disable, 22
 - interface additional metric configuration, 36
 - interface advertisement control, 20
 - interface reception control, 20
 - IPv6. *See* RIPng
 - maintaining, 30
 - neighbor specification, 27
 - network management configuration, 27
 - network optimization, 24
 - network tuning, 24
 - operation, 17
 - packet max length, 28
 - packet send rate configuration, 28
 - poison reverse configuration, 25
 - poison reverse enable, 25
 - preference configuration, 24
 - protocols and standards, 18
 - received/redistributed route filtering, 23
 - RIPv1 message zero field check enable, 26
 - RIPv2 message authentication configuration, 26
 - RIPv2 route summarization configuration, 21
 - route control configuration, 21
 - route entries, 17
 - route redistribution, 34
 - route redistribution configuration, 24
 - routing loop prevention, 17
 - split horizon configuration, 25
 - split horizon enable, 25
 - timer configuration, 24
 - update source IP address check, 26
 - version configuration, 20
 - versions, 18
- RIPng, 66, *See also* RIP
 - basic configuration, 67, 73
 - configuration, 66, 67, 73
 - default route advertisement, 69
 - displaying, 72
 - GR configuration, 71
 - IPsec profile application, 72
 - IPsec profile configuration, 78
 - maintaining, 72
 - network optimization, 70
 - network tuning, 70
 - packet, 66

- packet zero field check, 71
- poison reverse configuration, 70
- preference configuration, 69
- protocols and standards, 67
- received/redistributed route filtering, 69
- route control, 68
- route entry, 66
- route redistribution, 75
- route redistribution configuration, 70
- route summarization, 68
- routing metric configuration, 68
- split horizon configuration, 70
- timer configuration, 70
- RIPv1
 - message zero field check enable, 26
 - protocols and standards, 18
 - RIP basic configuration, 19
 - RIP configuration, 17, 18, 31
 - RIP versions, 18
 - version configuration, 20
- RIPv2
 - automatic route summarization enable, 22
 - message authentication configuration, 26
 - protocols and standards, 18
 - RIP basic configuration, 19
 - RIP configuration, 17, 18, 31
 - RIP versions, 18
 - route summarization configuration, 21
 - summary route advertisement, 22
 - version configuration, 20
- route
 - IP routing FIB route max lifetime, 3
 - IP routing FIB table optimal routes, 1
 - IP routing IPv6 default route configuration, 65
 - IP routing IPv6 static route BFD configuration, 55
 - IP routing IPv6 static route BFD control mode (direct next hop), 55
 - IP routing IPv6 static route BFD control mode (indirect next hop), 55
 - IP routing IPv6 static route BFD echo mode (single hop), 56
 - IP routing IPv6 static route configuration, 55
 - IP routing IPv6 static routing basic configuration, 57
 - IP routing IPv6 static routing BFD (direct next hop), 59
 - IP routing IPv6 static routing BFD (indirect next hop), 61
 - IP routing IPv6 static routing configuration, 55, 57
 - IP routing RIB label max lifetime, 3
 - IP routing RIB route max lifetime, 3
 - IP routing RIP default route advertisement, 23
 - IP routing RIP host route reception disable, 22
 - IP routing RIP poison reverse configuration, 25
 - IP routing RIP preference configuration, 24
 - IP routing RIP received/redistributed route filtering, 23
 - IP routing RIP route control configuration, 21
 - IP routing RIP route entries, 17
 - IP routing RIP route redistribution, 34
 - IP routing RIP route redistribution configuration, 24
 - IP routing RIP split horizon configuration, 25
 - IP routing RIP update source IP address check, 26
 - IP routing RIPng default route advertisement, 69
 - IP routing RIPng preference, 69
 - IP routing RIPng received/redistributed route filtering, 69
 - IP routing RIPng route control, 68
 - IP routing RIPng route entry, 66
 - IP routing RIPng route redistribution, 70, 75
 - IP routing RIPng route summarization, 68
 - IP routing RIPv1 message zero field check, 26
 - IP routing RIPv2 summary route advertisement, 22
 - IP routing route backup, 2
 - IP routing route preference, 2
 - IP routing route recursion, 2
 - IP routing route redistribution, 3
 - IP routing static configuration, 6
 - IP routing static default route, 16
 - IP routing static route, 6, 8
 - IP routing static route basics, 8
 - IP routing static route BFD, 6
 - IP routing static route BFD (direct next hop), 10
 - IP routing static route BFD (indirect next hop), 12
 - routing policy filters, 89
- routing
 - IP routing RIPng packet, 66
 - IP routing RIPng route entry, 66
 - IPv6 default route. *See* under IPv6 static routing IPv6 policy-based routing. *See* IPv6 PBR
 - IPv6 static routing. *See* IPv6 static routing policy-based routing. *Use* PBR
 - Routing Information Protocol. *Use* RIP
 - routing policy
 - display, 93
 - maintain, 93

S

security
IP routing RIPng IPsec profile application, 72
IP routing RIPng IPsec profile configuration, 78

setting
IP routing RIP packet max length, 28

source
IP routing RIP source IP address check, 26

specifying
IP routing RIP neighbor, 27

split horizon
configuration, 25
enable, 25
IP routing RIP routing loop prevention, 17
IP routing RIPng configuration, 70

static
routing. See [static routing](#)

static routing
basic configuration, 8
BFD configuration (direct next hop), 10
BFD configuration (indirect next hop), 12
configuration, 6, 8
default route configuration, 16
displaying, 8
IP routing static route BFD configuration, 6
IPv6. See [IPv6 static routing](#)
route configuration, 6
static route BFD bidirectional control mode (direct next hop), 7
static route BFD bidirectional control mode (indirect next hop), 7
static route BFD single-hop echo mode, 7

summarizing
IP routing RIPng route summarization, 68
IP routing RIPv2 automatic route summarization enable, 22
IP routing RIPv2 route summarization configuration, 21
IP routing RIPv2 summary route advertisement, 22

suppress timer (RIP), 24

T

table
IP routing, 1

timeout timer (RIP), 24

timer
IP routing RIP garbage-collect timer, 24
IP routing RIP suppress timer, 24
IP routing RIP timeout timer, 24
IP routing RIP update timer, 24
IP routing RIPng timer configuration, 70

topology
IP routing IPv6 default route configuration, 65
IP routing IPv6 static route configuration, 55
IP routing IPv6 static routing basic configuration, 57
IP routing IPv6 static routing BFD (direct next hop), 59
IP routing IPv6 static routing BFD (indirect next hop), 61
IP routing IPv6 static routing configuration, 55, 57

Track
IP routing IPv6 PBR collaboration, 82
IP routing PBR collaboration, 48
IP routing static route, 6

tuning
IP routing RIP networks, 24
IP routing RIPng network, 70

U

UDP
IP routing RIP configuration, 17, 18, 31
IP routing RIPng basic configuration, 67, 73
IP routing RIPng configuration, 66, 67, 73
IP routing RIPng GR configuration, 71
IP routing RIPng IPsec profile configuration, 78
IP routing RIPng route redistribution, 75

unicast
IP routing configuration, 1
IP routing dynamic routing protocols, 2
IP routing route backup, 2
IP routing route preference, 2
IP routing route recursion, 2
IP routing route redistribution, 3

update timer (RIP), 24

Z

zero field check
IP routing RIPng packet, 71
zero field check (RIPv1), 26