Contents

Configuring RIP

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

Overview

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 value range of the metric 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 route entries in a database. Each route 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 route 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 routing information on the interface from which the information was 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 so 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 discontiguous 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
Configuring basic RIP: • (Required.) Enabling RIP • (Optional.) Controlling RIP reception and advertisement on interfaces • (Optional.) Configuring a RIP version
(Optional.) Configuring RIP route control: • Configuring an additional routing metric • Configuring RIPv2 route summarization • Disabling host route reception • Advertising a default route • Configuring received/redistributed route filtering • Setting a preference for RIP • Configuring RIP route redistribution
(Optional.) Tuning and optimizing RIP networks: • Configuring RIP timers

Tasks at a glance
• Configuring split horizon and poison reverse • Setting the maximum number of ECMP routes • 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
(Optional.) Configuring RIP FRR

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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	By default, RIP is disabled.
3. Enable RIP on a network.	**network** *network-address* [*wildcard-mask*]	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
2. Enable RIP and enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	By default, RIP is disabled.
3. Return to system view.	**quit**	N/A
4. Enter interface view.	**interface** *interface-type interface-number*	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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Disable an interface from sending RIP messages.	**silent-interface** { *interface-type interface-number* | **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** *interface-type interface-number*	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 global nor interface-specific RIP version is configured, the interface sends RIPv1 broadcasts, and can receive RIPv1 broadcasts and unicasts, and RIPv2 broadcasts, multicasts, and unicasts.

To configure a RIP version:

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

Step	Command	Remarks
2. Enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Specify a global RIP version.	**version** { **1** | **2** }	By default, no global version is specified, and 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** *interface-type interface-number*	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, and 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** *interface-type interface-number*	N/A
3. Specify an inbound additional routing metric.	**rip metricin** [**route-policy** *route-policy-name*] *value*	The default setting is 0.
4. Specify an outbound additional routing metric.	**rip metricout** [**route-policy** *route-policy-name*] *value*	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*] [**vpn-instance** *vpn-instance-name*]	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 Ethernet 1/1 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*] [**vpn-instance** *vpn-instance-name*]	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*] [**vpn-instance** *vpn-instance-name*]	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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Enable RIP to advertise a default route.	**default-route** { **only** | **originate** } [**cost** *cost*]	By default, RIP does not advertise a default route.
4. Return to system view.	**quit**	N/A
5. Enter interface view.	**interface** *interface-type interface-number*	N/A
6. Configure the RIP interface to advertise a default route.	**rip default-route** { { **only** | **originate** } [**cost** *cost*] | **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 an IP prefix list. You can also configure RIP to receive routes only from a specified neighbor.

To configure route filtering:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A

Step	Command	Remarks
3. Configure the filtering of received routes.	**filter-policy** { *acl-number* | **gateway** *prefix-list-name* | **prefix-list** *prefix-list-name* [**gateway** *prefix-list-name*] } **import** [*interface-type interface-number*]	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** { *acl-number* | **prefix-list** *prefix-list-name* } **export** [*protocol* [*process-id*] | *interface-type interface-number*]	By default, the filtering of redistributed routes is not configured. This command filters redistributed routes, including routes redistributed with the **import-route** command.

Setting 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 set a preference for RIP:

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

Configuring RIP route redistribution

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

To configure RIP route redistribution:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Redistribute routes from another routing protocol.	**import-route** *protocol* [*process-id* | **all-processes** | **allow-ibgp**] [**cost** *cost* | **route-policy** *route-policy-name* | **tag** *tag*] *	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.) Set a default cost for redistributed routes.	**default cost** *value*	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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Configure RIP timers.	**timers** { **garbage-collect** *garbage-collect-value* | **suppress** *suppress-value* | **timeout** *timeout-value* | **update** *update-value* } *	By default: • 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** *interface-type interface-number*	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, but 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** *interface-type interface-number*	N/A
3. Enable poison reverse.	**rip poison-reverse**	By default, poison reverse is disabled.

Setting the maximum number of ECMP routes

Perform this task to implement load sharing over ECMP routes.

To set the maximum number of ECMP routes:

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Set the maximum number of ECMP routes.	**maximum load-balancing** *number*	By default, the maximum number of ECMP routes is 128.

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

Step	Command	Remarks
2. Enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	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 serial 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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	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** *interface-type interface-number*	N/A
3. Configure RIPv2 authentication.	**rip authentication-mode** { **md5** { **rfc2082** { **cipher** *cipher-string* | **plain** *plain-string* } *key-id* | **rfc2453** { **cipher** *cipher-string* | **plain** *plain-string* } } | **simple** { **cipher** *cipher-string* | **plain** *plain-string* } }	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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Specify a RIP neighbor.	**peer** *ip-address*	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** *process-id*	By default, MIB is bound to the RIP process with the smallest process ID.

Configuring the RIP packet sending rate

Perform this task to set 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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Set the interval for sending RIP packets and the maximum number of RIP packets that can be sent at each interval.	**output-delay** *time* **count** *count*	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** *interface-type interface-number*	N/A
3. Set the maximum length of RIP packets.	**rip max-packet-length** *value*	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") keep 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** [*process-id*] [**vpn-instance** *vpn-instance-name*]	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. This detection mechanism is not fast enough. 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** *ip-address*	By default, the source IP address of BFD echo packets is not configured.
3. Enter interface view.	**interface** *interface-type interface-number*	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 and the local device does 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** *ip-address*	By default, no source IP address is configured for BFD echo packets.
3. Enter interface view.	**interface** *interface-type interface-number*	N/A

Step	Command	Remarks
4. Enable BFD for RIP.	**rip bfd enable destination** *ip-address*	By default, BFD for RIP is disabled.

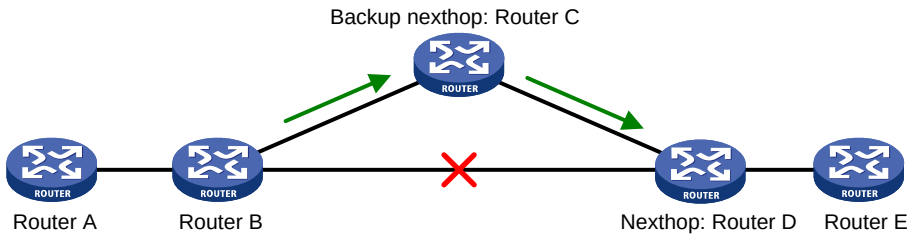
Configuring bidirectional control detection

Step	Command	Remarks
1. Enter system view.	**system-view**	N/A
2. Enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
3. Specify a RIP neighbor.	**peer** *ip-address*	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** *interface-type interface-number*	N/A
5. Enable BFD on the RIP interface.	**rip bfd enable**	By default, BFD is disabled on a RIP interface.

Configuring RIP FRR

A link or router failure on a path can cause packet loss and even routing loop until RIP completes routing convergence based on the new network topology. FRR uses BFD to detect failures and enables fast rerouting to minimize the impact of link or node failures.

Figure 1 Network diagram for RIP FRR



In Figure 1, configure FRR on Router B by using a routing policy to specify a backup next hop. When the primary link fails, RIP directs packets to the backup next hop. At the same time, RIP calculates the shortest path based on the new network topology, and forwards packets over that path after network convergence.

Configuration restrictions and guidelines

- RIP FRR takes effect only for RIP routes learned from directly connected neighbors.
- Do not use RIP FRR and BFD for RIP at the same time. Otherwise, FRR might fail to work.

- RIP FRR is available only when the state of primary link (with Layer 3 interfaces staying up) changes from bidirectional to unidirectional or down.

Configuration prerequisites

You must specify a next hop by using the **apply fast-reroute backup-interface** command in a routing policy and reference the routing policy for FRR. For more information about routing policy configuration, see "Configuring routing policies."

Configuration procedure

Configuring RIP FRR

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** *ip-address*	By default, the source IP address of BFD echo packets is not configured.
3. Enter RIP view.	**rip** [*process-id*] [**vpn-instance** *vpn-instance-name*]	N/A
4. Configure RIP FRR.	**fast-reroute route-policy** *route-policy-name*	By default, RIP FRR is disabled.

Enabling BFD for RIP FRR

By default, RIP FRR does not use BFD to detect primary link failures. To speed up RIP convergence, enable BFD single-hop echo detection for RIP FRR to detect primary link failures.

To configure BFD for RIP FRR:

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** *ip-address*	By default, the source IP address of BFD echo packets is not configured.
3. Enter interface view.	**interface** *interface-type interface-number*	N/A
4. Enable BFD for RIP FRR.	**rip primary-path-detect bfd echo**	By default, BFD for RIP FRR is disabled.

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** [*process-id*]
Display active routes in RIP database.	**display rip** *process-id* **database** [*ip-address* { *mask-length* | *mask* }]

Task	Command
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 of a RIP process.	**reset rip** *process-id* **statistics**

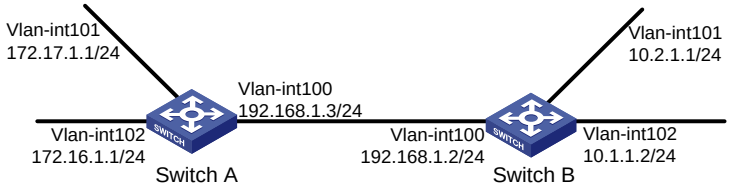
RIP configuration examples

Configuring basic RIP

Network requirements

As shown in Figure 2, 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 2 Network diagram



Configuration procedure

1. Configure IP addresses for interfaces. (Details not shown.)
2. Configure basic RIP by using either of the following methods:

   (Method 1) # Enable RIP on the specified networks on Switch A.

```
<SwitchA> system-view
[SwitchA] rip
[SwitchA-rip-1] network 192.168.1.0
[SwitchA-rip-1] network 172.16.0.0
[SwitchA-rip-1] network 172.17.0.0
[SwitchA-rip-1] quit
```

   (Method 2) # 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 192.168.1.2  on Vlan-interface100
      Destination/Mask        Nexthop           Cost    Tag     Flags   Sec
      10.0.0.0/8              192.168.1.2       1       0       RAOF    11
```

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 192.168.1.2 on Vlan-interface100
      Destination/Mask        Nexthop           Cost    Tag     Flags   Sec
      10.0.0.0/8              192.168.1.2       1       0       RAOF    50
      10.2.1.0/24             192.168.1.2       1       0       RAOF    16
      10.1.1.0/24             192.168.1.2       1       0       RAOF    16
```

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 192.168.1.3 on Vlan-interface100
      Destination/Mask        Nexthop           Cost    Tag     Flags   Sec
      172.16.1.0/24           192.168.1.3       1       0       RAOF    19
```

```
     172.17.1.0/24           192.168.1.3       1       0       RAOF    19
```

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 172.16.1.0 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 192.168.1.2 on Vlan-interface100
      Destination/Mask        Nexthop          Cost    Tag     Flags   Sec
      10.1.1.0/24             192.168.1.2      1       0       RAOF    19
```

   # 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 192.168.1.3 on Vlan-interface100
      Destination/Mask        Nexthop          Cost    Tag     Flags   Sec
      172.16.1.0/24           192.168.1.3      1       0       RAOF    19
```

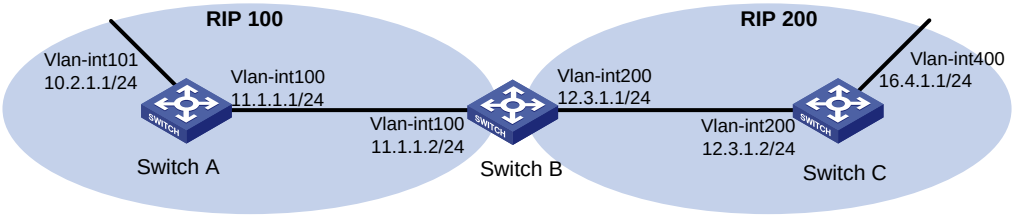
Configuring RIP route redistribution

Network requirements

As shown in Figure 3, 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 3 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
```

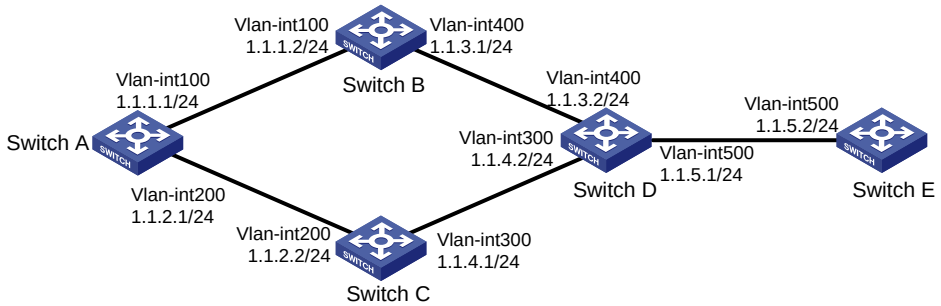
Configuring an additional metric for a RIP interface

Network requirements

As shown in Figure 4, 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 4 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
[SwitchB] 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, with the next hops as Switch B (1.1.1.2) and Switch C (1.1.2.2), and with the same cost of 2.

3. Configure an additional metric for a RIP interface:

Configure an 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.

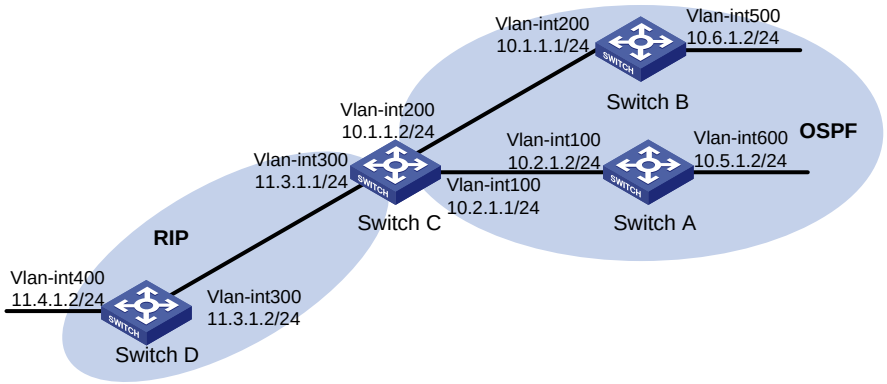
Configuring RIP to advertise a summary route

Network requirements

As shown in Figure 5, Switch A and Switch B run OSPF, Switch D runs RIP, and Switch C runs OSPF and RIP. Configure RIP to redistribute OSPF routes on Switch C so Switch D can learn routes destined for networks 10.1.1.0/24, 10.2.1.0/24, 10.5.1.0/24, and 10.6.1.0/24.

To reduce the routing table size of Switch D, configure route summarization on Switch C to advertise only the summary route 10.0.0.0/8 to Switch D.

Figure 5 Network diagram



Configuration procedure

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

   # Configure Switch A.

```
<SwitchA> system-view
[SwitchA] ospf
[SwitchA-ospf-1] area 0
[SwitchA-ospf-1-area-0.0.0.0] network 10.5.1.0 0.0.0.255
[SwitchA-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[SwitchA-ospf-1-area-0.0.0.0] quit
```

   # Configure Switch B.

```
<SwitchB> system-view
[SwitchB] ospf
[SwitchB-ospf-1] area 0
[SwitchB-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[SwitchB-ospf-1-area-0.0.0.0] network 10.6.1.0 0.0.0.255
[SwitchB-ospf-1-area-0.0.0.0] quit
```

Configure Switch C.

```
<SwitchC> system-view
[SwitchC] ospf
[SwitchC-ospf-1] area 0
[SwitchC-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[SwitchC-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[SwitchC-ospf-1-area-0.0.0.0] quit
[SwitchC-ospf-1] quit
```

3. Configure basic RIP:

Configure Switch C.

```
[SwitchC] rip 1
[SwitchC-rip-1] network 11.3.1.0
[SwitchC-rip-1] version 2
[SwitchC-rip-1] undo summary
```

Configure Switch D.

```
<SwitchD> system-view
[SwitchD] rip 1
[SwitchD-rip-1] network 11.0.0.0
[SwitchD-rip-1] version 2
[SwitchD-rip-1] undo summary
[SwitchD-rip-1] quit
```

Configure RIP to redistribute routes from OSPF process 1 and direct routes on Switch C.

```
[SwitchC-rip-1] import-route direct
[SwitchC-rip-1] import-route ospf 1
[SwitchC-rip-1] quit
```

Display the IP routing table on Switch D.

```
[SwitchD] 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
10.1.1.0/24         RIP    100  1            11.3.1.1        Vlan300
10.2.1.0/24         RIP    100  1            11.3.1.1        Vlan300
10.5.1.0/24         RIP    100  1            11.3.1.1        Vlan300
10.6.1.0/24         RIP    100  1            11.3.1.1        Vlan300
11.3.1.0/24         Direct 0    0            11.3.1.2        Vlan300
11.3.1.0/32         Direct 0    0            11.3.1.2        Vlan300
11.3.1.2/32         Direct 0    0            127.0.0.1       InLoop0
11.4.1.0/24         Direct 0    0            11.4.1.2        Vlan400
11.4.1.0/32         Direct 0    0            11.4.1.2        Vlan400
11.4.1.2/32         Direct 0    0            127.0.0.1       InLoop0
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
```

4. Configure route summarization:

Configure route summarization on Switch C and advertise only the summary route 10.0.0.0/8.

```
[SwitchC] interface vlan-interface 300
[SwitchC-Vlan-interface300] rip summary-address 10.0.0.0 8
```

Display the IP routing table on Switch D.

```
[SwitchD] display ip routing-table

Destinations : 12        Routes : 12

Destination/Mask    Proto  Pre  Cost         NextHop         Interface
0.0.0.0/32          Direct 0    0            127.0.0.1       InLoop0
10.0.0.0/8          RIP    100  1            11.3.1.1        Vlan300
11.3.1.0/24         Direct 0    0            11.3.1.2        Vlan300
11.3.1.0/32         Direct 0    0            11.3.1.2        Vlan300
11.3.1.2/32         Direct 0    0            127.0.0.1       InLoop0
11.4.1.0/24         Direct 0    0            11.4.1.2        Vlan400
11.4.1.0/32         Direct 0    0            11.4.1.2        Vlan400
11.4.1.2/32         Direct 0    0            127.0.0.1       InLoop0
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
```

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

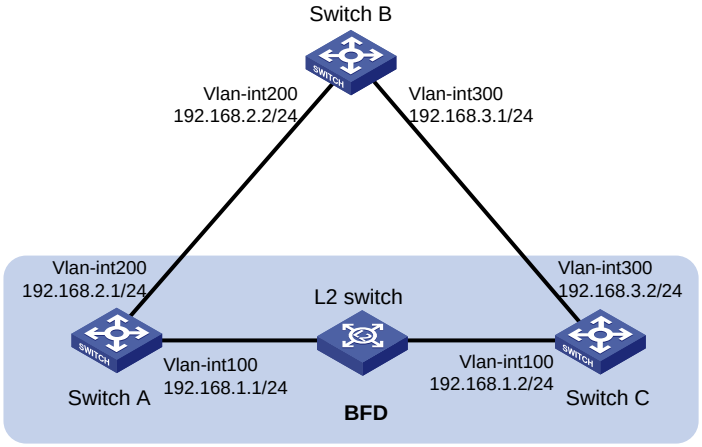
Network requirements

As shown in Figure 6, 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 so Switch A can 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 so RIP deletes the neighbor relationship and route information learned on VLAN-interface 100, and uses the route destined for 100.1.1.1 24 through VLAN-interface 200.

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] 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.

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

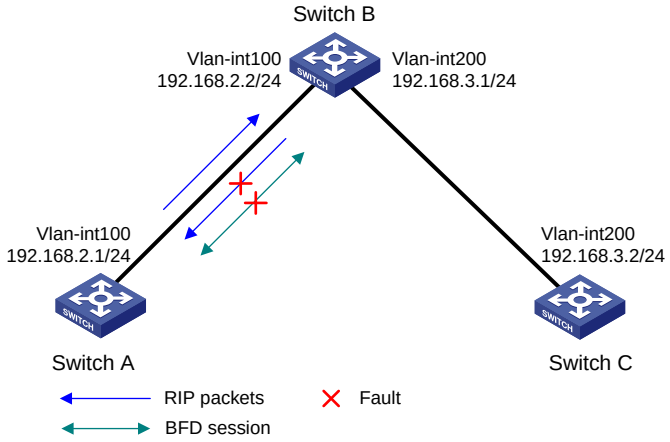
Network requirements

As shown in Figure 7, 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 so Switch B can 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 (packets from Switch A can reach Switch B, but packets from Switch B cannot reach Switch A), 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, and 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 7 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
```

Configuring BFD for RIP (bidirectional detection in BFD control packet mode)

Network requirements

As shown in Figure 8, 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, and enable static route redistribution into RIP on Switch A and Switch C so Switch A can learn two routes destined for 100.1.1.0/24 through VLAN-interface 100 and VLAN-interface 300, and uses the one 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 so RIP deletes the neighbor relationship and the route information received learned on VLAN-interface 100, and uses the route destined for 100.1.1.0/24 through VLAN-interface 300.

Figure 8 Network diagram

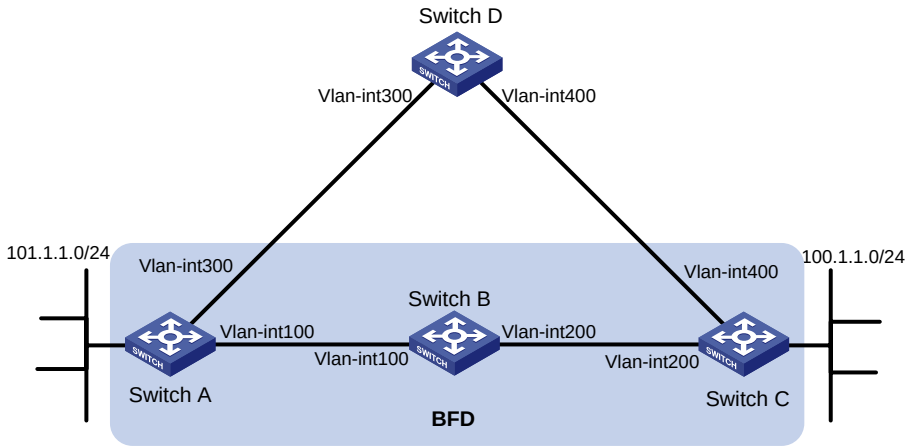


Table 1 Interface and IP address assignment

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.

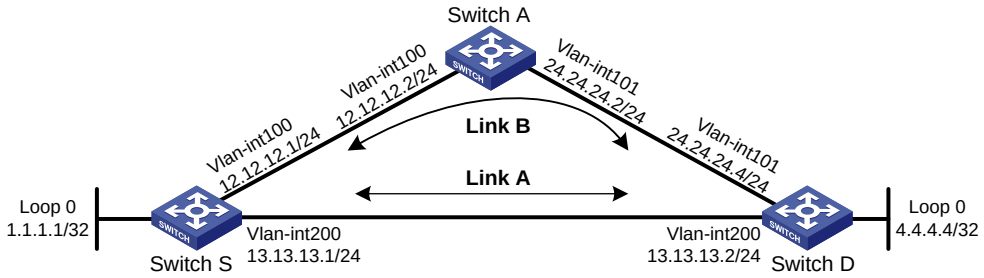
```
<SwitchA> display ip routing-table 100.1.1.0 24 verbose

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 RIP FRR

Network requirements

As shown in Figure 9, Switch S, Switch A, and Switch D run RIPv2. Configure RIP FRR so that when Link A becomes unidirectional, services can be switched to Link B immediately.

Figure 9 Network diagram



Configuration procedure

1. Configure IP addresses and subnet masks for interfaces on the switches. (Details not shown.)
2. Configure RIPv2 on the switches to make sure Switch A, Switch D, and Switch S can communicate with each other at Layer 3. (Details not shown.)
3. Configure RIP FRR:

   # Configure Switch S.

   ```
   <SwitchS> system-view
   [SwitchS] bfd echo-source-ip 2.2.2.2
   [SwitchS] ip prefix-list abc index 10 permit 4.4.4.4 32
   [SwitchS] route-policy frr permit node 10
   [SwitchS-route-policy-frr-10] if-match ip address prefix-list abc
   ```

```
[SwitchS-route-policy-frr-10] apply fast-reroute backup-interface vlan-interface
100 backup-nexthop 12.12.12.2
[SwitchS-route-policy-frr-10] quit
[SwitchS] rip 1
[SwitchS-rip-1] fast-reroute route-policy frr
[SwitchS-rip-1] quit
```

Configure Switch D.

```
<SwitchD> system-view
[SwitchD] bfd echo-source-ip 3.3.3.3
[SwitchD] ip prefix-list abc index 10 permit 1.1.1.1 32
[SwitchD] route-policy frr permit node 10
[SwitchD-route-policy-frr-10] if-match ip address prefix-list abc
[SwitchD-route-policy-frr-10] apply fast-reroute backup-interface vlan-interface
101 backup-nexthop 24.24.24.2
[SwitchD-route-policy-frr-10] quit
[SwitchD] rip 1
[SwitchD-rip-1] fast-reroute route-policy frr
[SwitchD-rip-1] quit
```

Verifying the configuration

Display route 4.4.4.4/32 on Switch S to view the backup next hop information.

```
[SwitchS] display ip routing-table 4.4.4.4 verbose

Summary Count : 1

Destination: 4.4.4.4/32
   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: 13.13.13.2
      Flags: 0x1008c        OrigNextHop: 13.13.13.2
      Label: NULL           RealNextHop: 13.13.13.2
    BkLabel: NULL             BkNextHop: 12.12.12.2
  Tunnel ID: Invalid          Interface: Vlan-interface200
BkTunnel ID: Invalid        BkInterface: Vlan-interface100
```

Display route 1.1.1.1/32 on Switch D to view the backup next hop information.

```
[SwitchD] display ip routing-table 1.1.1.1 verbose

Summary Count : 1

Destination: 1.1.1.1/32
   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: 13.13.13.1
      Flags: 0x1008c        OrigNextHop: 13.13.13.1
      Label: NULL           RealNextHop: 13.13.13.1
    BkLabel: NULL             BkNextHop: 24.24.24.2
  Tunnel ID: Invalid          Interface: Vlan-interface200
BkTunnel ID: Invalid        BkInterface: Vlan-interface101
```